

Dynamic Control in Stochastic Processing Networks

A Dissertation
Presented to
The Academic Faculty

by

Wuqin Lin

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Industrial and Systems Engineering
Georgia Institute of Technology
May 2005

Copyright © 2005 by Wuqin Lin

Dynamic Control in Stochastic Processing Networks

Approved by:

Dr. Jiangang Dai, Advisor
School of Industrial & Systems
Engineering
Georgia Institute of Technology

Dr. Hayriye Ayhan
School of Industrial & Systems
Engineering
Georgia Institute of Technology

Dr. Robert Foley
School of Industrial & Systems
Engineering
Georgia Institute of Technology

Dr. Anton Kleywegt
School of Industrial & Systems
Engineering
Georgia Institute of Technology

Dr. Amy Ward
School of Industrial & Systems
Engineering
Georgia Institute of Technology

Dr. Cathy Xia
IBM Research

Date Approved: May 4, 2005

ACKNOWLEDGEMENTS

I am grateful to my advisor, Professor Jim Dai, for his guidance, support, encouragement and patience, and especially for the inspiration that he provided for me throughout my PhD studies. I am deeply indebted to him for his great effort to sharpen my understanding and make the presentation of this dissertation clearer during the time when he most needed rest.

I thank Professor Hayriye Ayhan, Professor Robert Foley, Professor Anton Kleywegt, Professor Amy Ward, and Dr. Cathy Xia for their comments on my dissertation. I would like to especially thank Professor Ayhan and Professor Ward for their detailed comments on an early draft of this dissertation.

I want to thank Sasha Stolyar for his preprints on the Maxweight policy. His papers and Michael Harrison's recent papers on stochastic processing networks have inspired this study.

I am grateful to Professor Hong Chen and Professor Yat-Wah Wan for introducing me to stochastics, and to my master thesis advisor, Professor Raymond Cheung, for his guidance and introducing me to optimization.

I would like to thank Na An, Liwei Bai, Junxia Chang, Chao Chen, Keke Chen, Zesheng Chen, Brian Fralix, Jinxiang Gu, Yongpei Guan, Ping Jing, Matthew Jones, Zhaosong Lu, Yi Ma, Josh Reed, Jin Shi, Haibin Sun, Ni Wang, Zhendong Xia, Yun Xing, and Jiheng Zhang for their friendship. My academic brother, Jiheng Zhang, helped me understand Bramson's state space collapse framework. Thanks also go to Danilo Ardagna, Zhen Liu, Cathy Xia, and Li Zhang for their friendship and support during my internship at IBM research center.

I thank my father, my mother, and my sister for their emotional support and their faith in me. I thank my wife and best friend, Jihong, for her love and being with me all along.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	vi
SUMMARY	vii
CHAPTERS	
I INTRODUCTION	1
1.1 Notation	7
II STOCHASTIC PROCESSING NETWORKS	8
2.1 Resource Consumption	8
2.2 Routing	9
2.3 Resource Allocations	10
2.4 Service Policies	10
2.5 Stochastic Processing Network Equations	11
III THE MAXIMUM PRESSURE SERVICE POLICIES	13
IV STABILITY	16
4.1 The Static Planning Problem and the Main Stability Theorems	16
4.2 Fluid Models and an Outline of the Proof of Theorem 4.2	19
4.3 Fluid Limits	22
4.3.1 Fluid limits under a maximum pressure policy	25
4.4 EAA Assumption Revisited	27
4.4.1 Strict Leontief networks	28
4.4.2 A network example not satisfying the EAA assumption	28
4.5 Non-processor-splitting Service Policies	31
4.6 Non-preemptive Service Policies	36
4.7 Applications	43
4.7.1 Networks with alternate routing	44
4.7.2 Networks of data switches	47

V	ASYMPTOTIC OPTIMALITY	52
5.1	Workload Process and Complete Resource Pooling	53
5.2	Main Asymptotic Optimality Result	55
5.3	Outline of the Proof of Asymptotic Optimality	60
5.4	State Space Collapse for the Fluid Model	65
5.5	State Space Collapse	69
5.5.1	Probability estimates	71
5.5.2	State space collapse on $[\xi_{r,0}\tau_0/r^2, T]$	74
5.5.3	State space collapse on $[0, \xi_{r,0}L/r^2]$	80
5.5.4	Proof of Theorem 5.5	83
5.6	Proof of the Heavy Traffic Limit Theorem	87
VI	CONCLUSIONS AND FUTURE WORK	89
APPENDICES		
APPENDIX A	— EQUIVALENT DUAL STATIC PLANNING PROBLEM	91
APPENDIX B	— PROOFS	95
REFERENCES	102

LIST OF FIGURES

Figure 4.1 An example that does not satisfy the EAA assumption	29
Figure 4.2 A processing network for which processor sharing is important	33
Figure 4.3 Non-preemption could make unstable	37
Figure 4.4 A queueing network with alternate routes	44
Figure 4.5 A routing network by Laws and Louth	46
Figure 4.6 A network of input-queued switches	47

SUMMARY

Complex systems like semiconductor wafer fabrication facilities (fabs), networks of data switches, and large scale call centers all demand efficient resource allocation. Deterministic models like linear programs (LP) have been used for capacity planning at both the design and expansion stages of such systems. LP-based planning is critical in setting a medium range or long term goal for many systems. But it does not translate into a day-to-day operational policy that must deal with discreteness of jobs and the randomness of the processing environment.

A stochastic processing network, advanced by J. Michael Harrison, is a system that takes inputs of materials of various kinds and uses various processing resources to produce outputs of materials of various kinds. Such a network provides a powerful abstraction of a wide range of real world systems. It provides high-fidelity stochastic models in diverse economic sectors including manufacturing, service and information technology. The key goal of this research is to devise dynamic, operational policies that can achieve long term objectives for networks. These objectives include (i) achieving maximum throughput predicted by LPs, and furthermore, (ii) minimizing work-in-process, holding cost, or delay in networks.

In this research, we propose a family of operational policies called maximum pressure policies. The maximum pressure policies are attractive in that their implementation uses minimal state information of the network. The deployment of a resource (server) is decided based on the queue lengths in its serviceable buffers and the queue lengths in their immediately downstream buffers. In particular, the decision does not use arrival rate information that is often difficult or impossible to estimate reliably.

We prove that a maximum pressure policy can maximize throughput for a general class of stochastic processing networks. The fluid model approach is a powerful tool to prove whether an operational policy is throughput optimal for multiclass queueing networks. We extend this approach to more general stochastic processing networks and prove that a

maximum pressure policy is throughput optimal by showing that the fluid model under a maximum pressure policy is weakly stable.

We also establish an asymptotic optimality of maximum pressure policies for stochastic processing networks with a unique bottleneck. The optimality is in terms of stochastically minimizing *workload process*. We conduct the heavy traffic analysis for stochastic processing networks under maximum pressure policies. Bramson and Williams provided a powerful framework to prove heavy traffic limit theorems for multiclass queueing networks. We apply this framework to stochastic processing networks and prove a heavy traffic limit theorem for stochastic processing networks under maximum pressure policies. A key to the proof is to show that the network processes under maximum pressure policies exhibit a state space collapse.

CHAPTER I

INTRODUCTION

In a series of three papers, J. Michael Harrison [36, 37, 38] introduced progressively more general stochastic models, called *stochastic processing networks*. These networks are much more general than multiclass queueing networks that have been the subject of intensive study in the research community in the last 15 years. See, for example, [10, 12, 18, 21, 33, 40, 44, 55, 71, 75].

Loosely speaking, an open processing network is a system that takes inputs of materials of various kinds and uses various processing resources to produce outputs of materials of various (possibly different) kinds. Here, material is used as a generic substitute for a variety of entities that a system might process such as jobs, customers, packets, commodities, etc.; we use material and job interchangeably in the rest of this thesis. In fact, material need not be discrete although in this research we will focus on the case in which all jobs are discrete. Typically, there are constraints on the amount of material that a given server can process in a given time period. In addition, material may be processed by several servers, may be split up, or combined with other kinds of materials, before a final output is produced. Control is exerted through allocations of processors for the processing of one or more kinds of materials.

As observed in Bramson and Williams [16], deterministic (or average) models for describing such processing networks have a long history in economics and operations research. For example, the book “Activity Analysis of Production and Allocation” edited by T. C. Koopmans [43], provides an excellent summary of the early stages of development of such models. A prominent role is played there by the notion of a *processing activity*, which consumes certain kinds of materials, produces certain (possibly different) kinds of materials, and uses certain processors in the process. In a sense, Harrison’s [36] model of an open stochastic processing network is a stochastic analog of dynamic deterministic production

models such as those first considered by Dantzig [28].

In this research, we focus on a special class of Harrison's model. Even this specialized class of stochastic processing networks is broad enough to cover a wide range of application fields including manufacturing systems, service systems, computer systems, and computer communication networks. In addition to many systems that can be modeled by multiclass queueing networks, the added features of a stochastic processing network can model many new elements like machine-operator interaction and material handling in a semiconductor wafer fabrication facility, cross trained workers at a call center, networks of data switches, parallel processing computer systems, and routing in the Internet. Section 4.7 provides more detailed descriptions for some applications including networks of data switches and queueing networks with alternate routes. Readers are encouraged to jump to this section to get a feel of the scope of the application domains.

We are interested in the dynamic control of these stochastic processing networks at the operational level so that their long term objectives are met. Two types of long term objectives are often considered in the literature: (i) maximizing system throughput and (ii) minimizing work-in-process, holding cost, or delay of the system.

The maximum throughput or processing capacity of a system is often constrained by the processing speed of bottleneck resources. These constraints can be turned into a linear program (LP) from which the system processing capacity can be determined. This approach has been used in practice in capacity planning at either the design or expansion stage of a system. See, for example, Thomas and McClain [69]. LP based planning is very much relevant in setting a medium range or long term goal for the system. Since servers have overlapping processing capabilities, sometimes it may not be possible for any operational policy to achieve the maximum throughput predicted by the LP (see the example in Section 4.5). Indeed, even in the multiclass queueing network setting, it is now well known that many commonly used service policies including first-in-first-out are not throughput optimal (see Bramson [14] and Seidman [61]).

In our first line of research, we propose a family of operational policies called *maximum*

pressure policies. There are two versions of these policies, depending on whether processor-splitting is allowed in the policy space. We prove that both versions of these policies are throughput optimal under an extreme-allocation-available (EAA) assumption on the network structure. The assumption is satisfied for a wide class of networks. Such networks include multiclass queueing networks, parallel server systems, networks of data switches and queueing networks with alternate routes. In addition, we explicitly characterize, through linear programs, the stability regions of stochastic processing networks operating under a maximum pressure policy.

Our maximum pressure policies are generalizations of some forms of MaxWeight policies studied in Andrews et al. [2] and Stolyar [64] in the network setting. In their papers, the authors studied one-pass systems in which each job leaves the system after being processed at one processing step. Except for the network structure limitation, their works are actually more general than ours in the following two respects: (i) Job processing can depend on a stationary, random environment, and (ii) their MaxWeight policies are a richer family of policies for a one-pass system (see Section 3 for more detailed discussion). Although it has not been attempted here, it should be straightforward to generalize our results to stochastic processing networks with random environments. However, it is not at all clear how to generalize their general MaxWeight policies to our network setting.

Variants of maximum pressure and MaxWeight policies were first advanced by Tassiulas and Ephremides [67] under different names for scheduling a multihop radio network. Their work was further studied by various authors for systems in different applications [2, 27, 54, 64, 65, 66, 68], all limited to one-pass systems except [66]. The work of Tassiulas and Bhattacharya [66] represents a significant advance in finding efficient operational policies for a wide class of networks, and is closely related to our current work. Although we were ignorant of their work when our work was performed, there is a significant amount of overlap and difference between these two works. The following are the major contrasts of these two works. (a) They model server interdependence by directly imposing constraints on servers, whereas we use processing activities and constraints on them to model server interdependence; the latter is a more general approach to model server interdependence. (b)

Their model, when translated into our stochastic processing network framework, appears to be a special network within a class of *strictly unitary networks*. In a latter network, each activity requires a *single server* that processes jobs in a *single buffer*. Thus, their model cannot model activities that require simultaneous possession of multiple servers nor activities that can simultaneously process jobs from multiple buffers. In particular, we are not able to see how their model can model operator-machine interactions and networks of data switches (see Section 4.7.2). (c) Their model requires processing speeds to depend only on buffers, not on activities. This assumption rules out many models like skill-based routing in call-center environments. (d) Our exogenous arrival model is more general. This generality allows us to model alternate routes at source levels (see Section 4.7.1). As a consequence of these model differences, they can focus on non-processor-splitting, non-preemptive policies without additional assumptions on network structures. To prove maximum pressure policies are throughput optimal for our general model, we need to allow processor-splitting and preemption in our policies and to search for new assumptions on network structure (see EAA assumption in Section 4.4). (e) Only throughput optimality has been proven in [66]. No analysis of secondary performance measures was conducted. In our research, we prove that maximum pressure policies can asymptotically minimize the system workload for the stochastic processing networks with a unique bottleneck.

The maximum pressure policies are attractive in that their implementation uses minimal state information of the network. The deployment of a processor is decided based on the queue lengths in its serviceable buffers and the queue lengths at their immediately downstream buffers. In particular, the decision does not use arrival rate information that is often hard or impossible to estimate reliably. The maximum pressure policies are not completely local in that they use immediately downstream buffer information of a processor. Using such information is not an issue in many manufacturing systems, but may be a problem for other systems. Searching for a purely local policy that is throughput optimal remains an open problem. See Section 4.7.1 for more discussions on this point.

It is a more challenging problem to design dynamic control policies for stochastic processing networks that are simple to implement and yet are at least approximately optimal in

an appropriate sense in terms of some second order performance measures. As one approach to this problem, Harrison [33] proposed Brownian network models as the heavy traffic approximation to the stochastic processing networks. By analyzing Brownian network models and cleverly interpreting their solutions as control policies to the original stochastic networks, various researchers successfully developed good policies for some particular networks. But in general, Brownian models are not always tractable, and it is not easy to interpret the solutions as control policies for the original networks. There are very few proofs of asymptotic optimality of the interpreted policies even when the interpretation is possible. Most policies developed through this approach require arrival rate information.

A key step in Harrison’s approach is to form an *equivalent workload formulation* of the Brownian network model, explained in Harrison and Van Mieghem [34], by replacing the queue length process with a lower dimensional *workload process*. The Brownian model often has a simple solution when the workload process is one-dimensional. This corresponds to a complete resource pooling condition for the original stochastic processing network. Roughly speaking, the complete resource pooling condition requires enough overlap in the processing capacities of bottleneck servers such that these servers form a single pooled resource or “super server”. The complete resource pooling condition is articulated by the dual problem of a linear program (LP) called the *static planning problem*. For a network satisfying the complete resource pooling condition, the corresponding dual LP has a unique optimal solution and the one-dimensional workload process is defined by this unique optimal solution. More specifically, let (y, z) be the unique optimal solution to the dual problem. Then y_i is interpreted as the workload contribution per buffer i job to the pooled bottleneck resource. The workload process is defined as $W = y \cdot Z$, where $Z = \{Z(t), t \geq 0\}$ is the queue length process.

Our second line of research is to establish an asymptotic optimality of maximum pressure policies in terms of minimizing the workload process for stochastic processing networks with a unique bottleneck resource pool. Minimizing the workload process is important even if the ultimate objective is to minimize the delay or some type of holding cost [6, 8, 36, 64].

In this line of research, we focus on the networks that satisfy a heavy traffic condition

that at least one server has to be 100% busy in order to handle all the input. Our definition of heavy traffic condition is less restrictive than those in [6, 39] the authors require the network to be balanced; that is, every server in the network is heavily loaded. This balanced load requirement, combined with the complete resource pooling assumption, rules out some well known networks such as multiclass queueing networks. It is commonly believed that under heavy traffic scaling, non-bottleneck stations disappear in the limit. Thus, one can confine heavy traffic analysis to a subnetwork obtained by deleting all the non-bottleneck stations from the network. However, operational policies based on the bottleneck subnetwork have no natural extension to the original network. We will show that under a maximum pressure policy, non-bottleneck stations will disappear in heavy traffic limits. Such disappearance of non-bottleneck stations can by no means be assumed. This fact is false under some commonly used operational policies like first-in-first-out [15].

Our asymptotic optimality result, to some extent, greatly generalizes the results in Stolyar [64] from a one-pass system to the network setting. In [64], the author proved that MaxWeight policies asymptotically minimize the workload processes in heavy traffic for one-pass systems in which each job leaves the system after being processed at one processing step. As a generalization of MaxWeight policies, Mandelbaum and Stolyar [49] proposed generalized $c\mu$ rule and proved its asymptotic optimality for a general switch model. Parallel server systems were studied by Harrison [35], Harrison and Lopez [39], Bell and Williams [9, 8], and Williams [76]. Discrete review policies [35, 39] and continuous review threshold policies [9, 8, 76] were proposed to minimize the expected discounted cumulative linear holding costs. Ata and Kumar [6] recently proposed a discrete review policy to for a class of stochastic processing networks called *unitary networks*. The network model that we consider in our research is much more general than those that have been studied in the literature.

Bramson [12] and Williams [75] developed a general framework for proving state space collapse and heavy traffic limit theorems for multiclass queueing networks. Our proof of the main results is an example of extending the Bramson-Williams framework to more general stochastic processing networks. We first show that the fluid limits under maximum

pressure policies exhibit some form of state space collapse. Then we translate the state space collapse to the diffusion scaling using Bramson's approach [12]. Finally, the state space collapse is converted to a heavy traffic limit theorem, which immediately implies the asymptotic optimality.

The remainder of this dissertation is organized as follows. In Chapter 2, we introduce the stochastic processing networks to be studied in this research. The maximum pressure policies are defined in Chapter 3. In Chapter 4, we prove that the maximum pressure policies are throughput optimal. We establish an asymptotic optimality of the maximum pressure policies in Chapter 5. Chapter 6 contains some concluding remarks and directions of future research.

1.1 *Notation*

We denote the set of natural numbers as \mathbb{N} , and the set of nonnegative integer numbers as \mathbb{Z}^+ . We use \mathbb{R}^d to denote the d -dimensional Euclidean space. Vectors in \mathbb{R}^d will be column vectors unless indicated otherwise, and the transpose of a vector v will be denoted as v' . For $v, w \in \mathbb{R}^d$, $v \cdot w$ denotes the dot product. The max norm in \mathbb{R}^d is denoted as $|\cdot|$, and for a matrix A , we use $|A|$ to denote the maximum absolute value among all components. The product norm $\|\cdot\|$ in \mathbb{R}^d is defined by $\|v\| = \sqrt{v \cdot v}$. For $r_1, r_2 \in \mathbb{R}$, we denote $r_1 \vee r_2$ and $r_1 \wedge r_2$ to be respectively the maximum and minimum of r_1 and r_2 .

We shall use $\mathbb{D}^d[0, \infty)$ to denote the set of functions $f : [0, \infty) \mapsto \mathbb{R}^d$ that are right continuous on $[0, \infty)$ having left limits on $(0, \infty)$. For $f \in \mathbb{D}^d[0, \infty)$, we let

$$\|f\|_t = \sup_{0 \leq s \leq t} |f(s)|.$$

We endow the function space $\mathbb{D}^d[0, \infty)$ with the usual Skorohod J_1 -topology. A sequence $\{f_r\} \subset \mathbb{D}^d[0, \infty)$ is said to converge to an $f \in \mathbb{D}^d[0, \infty)$ uniformly on compact (u.o.c.) sets, denoted as $f_r(\cdot) \rightarrow f(\cdot)$, if for each $t \geq 0$, $\lim_{r \rightarrow \infty} \|f_r(s) - f(s)\|_t = 0$. We use " \Rightarrow " to denote convergence in distribution.

CHAPTER II

STOCHASTIC PROCESSING NETWORKS

In this section, we describe a variant of the class of stochastic processing networks advanced in Harrison [36]. The network is assumed to have $\mathbf{I} + 1$ buffers, \mathbf{J} activities and \mathbf{K} processors. Buffers, activities and processors are indexed by $i = 0, \dots, \mathbf{I}$, $j = 1, \dots, \mathbf{J}$ and $k = 1, \dots, \mathbf{K}$, respectively. For notational convenience, we define $\mathcal{I} = \{1, \dots, \mathbf{I}\}$ the set of buffers excluding buffer 0, $\mathcal{J} = \{1, \dots, \mathbf{J}\}$ the set of activities and $\mathcal{K} = \{1, \dots, \mathbf{K}\}$ the set of processors. Each buffer, with infinite capacity, holds jobs or materials that await service. Buffer 0 is a special one that is used to model the outside world, where an infinite number of jobs await. Each activity can simultaneously process jobs from a set of buffers. It may require simultaneous possession of multiple processors to be active. Jobs departing from a buffer will go next to other buffers with certain probabilities that depend on the current activity taken.

2.1 Resource Consumption

Each activity needs one or more processors available to be active. For activity j , $A_{kj} = 1$, if activity j requires processor k , and $A_{kj} = 0$ otherwise. The $\mathbf{K} \times \mathbf{J}$ matrix $A = (A_{kj})$ is the resource consumption matrix. Each activity may be allowed to process jobs in multiple buffers simultaneously. For activity j , we use the indicator function B_{ji} to record whether buffer i can be processed by activity j . ($B_{ji} = 1$ if activity j processes buffer i job.) The set of buffers i with $B_{ji} = 1$ is said to be the *constituency* of activity j . It is denoted by \mathcal{B}_j . The constituency is assumed to be nonempty for each activity $j \in \mathcal{J}$, and may contain more than one buffer. When a processing requirement of an activity is met, a job departs from each one of the constituent buffers. For each activity j , we use $u_j(\ell)/\mu_j$ to denote the ℓ th activity j processing requirement, where $u_j = \{u_j(\ell), \ell \geq 1\}$ is an i.i.d. sequence of random variables and μ_j is a strictly positive real number. We set $\sigma_j^2 = \text{var}(u_j(1))$, and assume that $\sigma_j < \infty$ and u_j is *unitized*, that is, $\mathbb{E}[u_j(1)] = 1$. It follows that $1/\mu_j$ and σ_j

are the mean and coefficient of variation, respectively, for the processing times of activity j .

An activity j is said to be an *input activity* if it processes jobs only from buffer 0, i.e., $\mathcal{B}_j = \{0\}$. An activity j is said to be a *service activity* if it does not process any job from buffer 0, i.e., $0 \notin \mathcal{B}_j$. We assume that each activity is either an input activity or a service activity. We further assume that each processor processes either input activities only or service activities only. A processor that only processes input activities is called an *input processor*, and a processor that only processes service activities is called an *service processor*. The input processors process jobs from buffer 0 (outside) and generate the arrivals for the network. We denote \mathcal{J}_I to be the set of input activities, \mathcal{J}_S the set of service activities, \mathcal{K}_I the set of input processors, and \mathcal{K}_S the set of service processors.

2.2 Routing

Buffer i jobs, after being processed by activity j , will go next to other buffers or leave the system. Let e_0 be the \mathbf{I} -dimensional vector of all 0's, and for $i \in \mathcal{I}$, e_i is the \mathbf{I} -dimensional vector with i th component 1 and other components 0. For each activity $j \in \mathcal{J}$ and each constituent buffer $i \in \mathcal{B}_j$, we use an \mathbf{I} -dimensional binary random vector $\phi_i^j(\ell) = (\phi_{ii'}^j(\ell), i' \in \mathcal{I})$ to denote the *routing vector* of the ℓ -th buffer i job processed by activity j , where $\phi_i^j(\ell) = e_{i'}$ if the ℓ -th buffer i job processed by activity j goes next to buffer i' , and $\phi_i^j(\ell) = e_0$ if the job leaves the system. We assume the sequence $\phi_i^j = \{\phi_i^j(\ell), \ell \geq 1\}$ is i.i.d. for each activity $j \in \mathcal{J}$ and $i \in \mathcal{B}_j$. Set $P_{ii'}^j = \mathbb{E}[\phi_{ii'}^j(1)]$, then $P_{ii'}^j$ is the probability that a buffer i job processed by activity j will go next to buffer i' .

For each $j \in \mathcal{J}, i \in \mathcal{B}_j$, the cumulative routing process is defined by the sum

$$\Phi_i^j(\ell) = \sum_{n=1}^{\ell} \phi_i^j(n),$$

and $\Phi_{ii'}^j(\ell)$ denotes the number of jobs that will go next to buffer i' among the first ℓ buffer i jobs that are processed by activity j .

The sequences

$$(u_j, \phi_i^j : i \in \mathcal{B}_j, j \in \mathcal{J}) \tag{2.1}$$

are said to be the *primitive increments* of the network. We assume that they are mutually

independent and all are independent of the initial state of the network.

2.3 Resource Allocations

Because multiple activities may require usage of the same processor, not all activities can be simultaneously undertaken at 100% level. For most of this thesis, we assume that each processor's service capacity is infinitely divisible, and processor-splitting of a processor's service capacity is realizable. We use a nonnegative variable a_j to denote the level at which processing activity j is undertaken. When $a_j = 1$, activity j is employed at a 100% level. When $a_j = 0$, activity j is not employed. Suppose that the engagement level of activity j is a_j , with $0 \leq a_j \leq 1$. The processing requirement of an activity j job is depleted at rate a_j . (The job finishes processing when its processing requirement reaches 0.) The activity consumes $a_j A_{kj}$ fraction of processor k 's service capacity per unit time. The remaining service capacity, $1 - a_j A_{kj}$, can be used for other activities.

We use $a = (a_j) \in \mathbb{R}_+^{\mathbf{J}}$ to denote the corresponding \mathbf{J} -dimensional allocation (column) vector, where \mathbb{R}_+ denotes the set of nonnegative real numbers. Since each processor k can decrease processing requirements at the rate of at most 1 per unit of time, we have

$$\sum_{j \in \mathcal{J}} A_{kj} a_j \leq 1 \quad \text{for each processor } k. \quad (2.2)$$

In vector form, $Aa \leq e$, where e is the \mathbf{K} -dimensional vector of ones. We assume that there is at least one input activity and that the input processors are never idle. Namely,

$$\sum_{j \in \mathcal{J}} A_{kj} a_j = 1 \quad \text{for each input processor } k. \quad (2.3)$$

We use \mathcal{A} to denote the set of allocations $a \in \mathbb{R}_+^{\mathbf{J}}$ that satisfy (2.2) and (2.3).

Each $a \in \mathcal{A}$ represents an allowable allocation of the processors working on various activities. We note that \mathcal{A} is bounded and convex. Let $\mathcal{E} = \{a^1, \dots, a^{\mathbf{E}}\}$ be the set of extreme points of \mathcal{A} , where the total number \mathbf{E} of extreme points is finite.

2.4 Service Policies

Each job in a buffer is assumed to be processed by one activity in its entire stay at the buffer. A processing of an activity can be preempted. In this case, each in-service job is "frozen"

by the activity. When the next time the activity is made active again, the processing is resumed from where it was left off. In addition to the availability of processors, a (non-preempted) activity can be made active only when each constituent buffer has jobs that are not in service nor frozen. We assume that within each buffer *head-of-line* policy is used. When a (non-preempted) activity becomes active with a given engagement level, the leading job in each buffer that is not in service nor frozen is processed. If multiple activities are actively working on a buffer, there are multiple jobs in the buffer that are in service. For an allocation a , if there is an activity j with $a_j > 0$ that cannot be made active, the allocation is infeasible. At any given time t , we use $\mathcal{A}(t)$ to denote the set of allocations that are *feasible* at that time. A policy specifies which allocation being undertaken at each time $t \geq 0$, and it is denoted as $\pi = \{\pi(t) : t \geq 0\}$. Under the policy π , allocation $\pi(t) \in \mathcal{A}(t)$ will be employed at time t .

2.5 Stochastic Processing Network Equations

For each activity j , we first define the counting process $S_j = \{S_j(t), t \geq 0\}$ associated with the processing requirement sequence $\{u_j(\ell)/\mu_j, \ell \geq 0\}$. For each $t \geq 0$,

$$S_j(t) = \max \left\{ n : \sum_{\ell=1}^n u_j(\ell) \leq \mu_j t \right\}. \quad (2.4)$$

Since the service policy is assumed to be head-of-line, $S_j(t)$ is the number of activity j processing completions in t units of activity j processing time. Note that a unit of *activity j processing time* is not the same as a unit of *activity j busy time*. When the activity is employed at level a_j , one unit of activity j busy time is equal to a_j units of activity j processing time. We use $T_j(t)$ to denote the cumulative activity j processing time in $[0, t]$. Let $T(t)$ be the corresponding \mathbf{J} -dimensional vector; we refer to $T = \{T(t), t \geq 0\}$ as the *cumulative activity level process*. For each buffer $i \in \mathcal{I}$, let buffer level $Z_i(t)$ denote the number of jobs in buffer i at time t . We use $Z(t)$ to denote the corresponding \mathbf{I} -dimensional column vector; we refer to $Z = \{Z(t), t \geq 0\}$ as the buffer level process. Denoting $\mathbb{X}(t) = (Z(t), T(t))$ for $t \geq 0$, we call $\mathbb{X} = \{\mathbb{X}(t) : t \geq 0\}$ the stochastic processing network process.

Now we can write down the equations describing the dynamics of the stochastic processing network:

$$Z_i(t) = Z_i(0) + \sum_{j \in \mathcal{J}} \sum_{i' \in \mathcal{B}_j} \Phi_{i'i}^j(S_j(T_j(t))) - \sum_{j \in \mathcal{J}} S_j(T_j(t)) B_{ji} \text{ for each } t \geq 0 \text{ and } i \in \mathcal{I}, \quad (2.5)$$

$$Z_i(t) \geq 0 \text{ for each } t \geq 0 \text{ and } i \in \mathcal{I}, \quad (2.6)$$

$$\sum_{j \in \mathcal{J}} A_{kj}(T_j(t) - T_j(s)) = t - s \text{ for each } 0 \leq s \leq t \text{ and each input processor } k, \quad (2.7)$$

$$\sum_{j \in \mathcal{J}} A_{kj}(T_j(t) - T_j(s)) \leq t - s \text{ for each } 0 \leq s \leq t \text{ and each processor } k, \quad (2.8)$$

$$T \text{ is nondecreasing and } T(0) = 0. \quad (2.9)$$

Since quantity $T_j(t)$ is the cumulative amount of activity j processing time in $[0, t]$, $S_j(T_j(t))$ is the number of activity j processings completed by time t , and $\sum_{j \in \mathcal{J}} S_j(T_j(t)) B_{ji}$ is the total number of jobs that depart from buffer $i \in \mathcal{I} \cup \{0\}$ in $[0, t]$. For each activity j , $\sum_{i' \in \mathcal{B}_j} \Phi_{i'i}^j(S_j(T_j(t)))$ is the total number of jobs sent to buffer i by activity j from its constituent buffers by time t , so $\sum_{j \in \mathcal{J}} \sum_{i' \in \mathcal{B}_j} \Phi_{i'i}^j(S_j(T_j(t)))$ is the total number of jobs that go to buffer i by time t . Equation (2.5) is the flow balance equation. It says that the number of jobs in buffer i at time t equals the initial number plus the number of arrivals subtracted from the number of departures. Inequality (2.8) holds because each processor can spend at most $t - s$ units of time processing various activities from time s to t , and equation (2.7) holds because each input processor is assumed to never idle. To uniquely define the network dynamics, it is enough to specify $T(t)$.

We note that equations (2.5)–(2.9) hold under any head-of-line service policy. They are called *stochastic processing network equations*. Under a specific service policy like a maximum pressure policy, there are additional equations for the network processes.

CHAPTER III

THE MAXIMUM PRESSURE SERVICE POLICIES

In this section, we define a family of policies, called *maximum pressure service policies*. Under a mild assumption on network structure, we will prove in Section 4.2 that a maximum pressure policy is throughput optimal in the sense that it stabilizes a stochastic processing network if the network is stabilizable at all.

For each buffer $i = 1, \dots, \mathbf{I}$ and each activity $j = 1, \dots, \mathbf{J}$ we define

$$R_{ij} = \mu_j \left(B_{ji} - \sum_{i' \in \mathcal{B}_j} P_{ii'}^j \right). \quad (3.1)$$

The $\mathbf{I} \times \mathbf{J}$ matrix $R = (R_{ij})$ is called the input-output matrix in Harrison [37]. (Harrison took R as part of a model specification to allow more general modeling capability.) One interprets R_{ij} as the average amount of buffer i material consumed per unit of activity j , with a negative value being interpreted to mean that activity j is a net producer of material in buffer i . For a resource allocation $a \in \mathcal{A}$ and a given $z \in \mathbb{R}_+^{\mathbf{I}}$, define the corresponding total network pressure to be

$$p(a, z) = z \cdot Ra, \quad (3.2)$$

where, for two vectors x and y , $x \cdot y = \sum_{\ell} x_{\ell} y_{\ell}$ denotes the inner product. Although we interpret z as the buffer level vector in the network, its components do not have to be integers.

The network pressure $p(a, z)$ can also be written as

$$p(a, z) = \sum_{j \in \mathcal{J}} a_j \mu_j \sum_{i \in \mathcal{B}_j} \left(z_i - \sum_{i' \in \mathcal{I}} P_{ii'}^j z_{i'} \right).$$

For each activity j , $\mu_j \sum_{i \in \mathcal{B}_j} (z_i - \sum_{i' \in \mathcal{I}} P_{ii'}^j z_{i'})$ is the pressure of activity j . It equals the processing rate times the buffer level difference between the service buffers and their immediate downstream buffers.

The general idea of a maximum pressure policy is to employ an allocation

$$a^* \in \operatorname{argmax}_{a \in \mathcal{A}} p(a, Z(t)) \quad (3.3)$$

at any given time t . Unfortunately, such an a^* is not always a feasible allocation. Note that $p(a, Z(t))$ is linear in a . Thus, the maximum in (3.3) is achieved at one of those extreme allocations. Namely, $p(a^*, Z(t)) = \max_{a \in \mathcal{E}} p(a, Z(t))$, where, as before, \mathcal{E} is the set of extreme allocations of \mathcal{A} .

Recall that $\mathcal{A}(t)$ is the set of feasible allocations at time t . Namely, $\mathcal{A}(t)$ is the set of allocations $a = (a_j)$ such that at time t for each non-preempted activity j with $a_j > 0$, the constituent buffers (those buffers i with $B_{ji} = 1$) have “fresh” jobs that are neither in service nor preempted. Define $\mathcal{E}(t) = \mathcal{E} \cap \mathcal{A}(t)$ to be the set of feasible extreme allocations at time t . Because preemption is assumed, one can argue that $\mathcal{E}(t)$ is always nonempty. For example, any extreme allocation that forces all service processors to stay idle is an element in $\mathcal{E}(t)$.

Definition 3.1. A service policy is said to be a *maximum pressure policy* if at each time t , the network chooses an allocation $a^* \in \operatorname{argmax}_{a \in \mathcal{E}(t)} p(a, Z(t))$.

When more than one allocation attain the maximum pressure, a tie-breaking rule is used. Our results are not affected by how ties are broken. However, for concreteness, one can order the extreme allocation set \mathcal{E} , and always chooses the smallest, maximum-pressure allocation.

Note the buffer level process Z does not change between processing completions. Thus, under a maximum pressure policy, allocations will not change between these completions. Each allocation decision is triggered by the completion of either an input activity or a service activity. Let $t_0 = 0$, and $\{t_n : n = 1, \dots\}$ be the sequence of decision times under a maximum pressure policy. At decision time t_n , one observes the buffer level $Z(t_n)$, and chooses an allocation $a^n = f(Z(t_n))$, where $f : \mathbb{R}_+^{\mathbf{I}} \rightarrow \mathcal{E} \subset \mathcal{A}$ is a function such that $f(z)$ maximizes $p(a, z)$ among all feasible allocations $a \in \mathcal{E}$ for each $z \in \mathbb{R}_+^{\mathbf{I}}$. The allocation remains fixed until the next activity completion time t_{n+1} .

For one-pass systems with no alternate routing, $R_{ij} = \mu_j B_{ji}$ and the maximum pressure policy is reduced to a special case of the MaxWeight policy proposed in [64]. In fact, MaxWeight policy is to employ an allocation $a \in \operatorname{argmax}_a \sum_i \sum_j C'_i(Z_i(t)) R_{ij} a_j$ where $C_i(\cdot)$ is any convex function. Setting $C_i(Z_i) = Z_i^2$, MaxWeight policy reduces to a maximum pressure policy.

CHAPTER IV

STABILITY

In this chapter, we study the stability of stochastic processing networks. We prove that, under a mild assumption on the network structure, a maximum pressure policy is throughput optimal in the sense that it stabilizes a stochastic processing network if the network is stabilizable at all.

We first define the pathwise stability for stochastic processing networks.

Definition 4.1. A stochastic processing network operating under a general service policy is said to be *pathwise stable* or simply *stable* if for every initial state, with probability one,

$$\lim_{t \rightarrow \infty} Z_i(t)/t = 0, \quad i \in \mathcal{I}. \quad (4.1)$$

Pathwise stability ensures that the total departure rate from the network is equal to the total input rate to the network. An unstable network incurs linear build up of jobs in the system, at least for some network realizations. Although it will not be discussed further in this thesis, one can employ other definitions of stability like positive Harris recurrence under some stronger assumptions on the primitive processes. Readers are referred to Dai [21], Dai and Meyn [26] and Stolyar [63] for such possible extensions.

The central focus of this chapter is to answer the following questions: (i) what is the natural condition on the primitive processes under which the stochastic processing network is stabilizable under some service policy? (ii) given that the network is stabilizable, are there any dynamic policies that use “minimal system information” and stabilize the network?

4.1 The Static Planning Problem and the Main Stability Theorems

In this section, we state the main stability theorems. We first introduce an LP called the static planning problem that will be used in the theorems to characterize the stability of

a stochastic processing network. For a stochastic processing network with input-output matrix R and capacity consumption matrix A , the static planning problem is defined as follows: choose a scalar ρ and a \mathbf{J} -dimensional column vector x so as to

$$\text{minimize} \quad \rho \quad (4.2)$$

$$\text{subject to} \quad Rx = 0, \quad (4.3)$$

$$\sum_{j \in \mathcal{J}} A_{kj} x_j = 1 \text{ for each input processor } k, \quad (4.4)$$

$$\sum_{j \in \mathcal{J}} A_{kj} x_j \leq \rho \text{ for each service processor } k, \quad (4.5)$$

$$x \geq 0. \quad (4.6)$$

For each optimal solution (ρ, x) to (4.2)–(4.6), the vector x is said to be a *processing plan* for the stochastic processing network, where component x_j is interpreted as the long-run fraction of time that activity j is undertaken. Since one of the constraints in (4.5) must be bounding for a service processor, ρ is interpreted as the long-run utilization of the busiest service processor under the processing plan. With this interpretation, the left side of (4.3) is interpreted as the long-run *net flow rates* from the buffers. Equality (4.3) demands that, for each buffer, the long-run input rate to the buffer is equal to the long-run output rate from the buffer. Equality (4.4) ensures that input processors are never idle, while inequality (4.5) requires that each service processor's utilization not exceed that of the busiest service processor. The objective is to minimize the utilization of the busiest service processor. For future references, the optimal objective value ρ is said to be the *traffic intensity* of the stochastic processing network.

The following theorem provides a partial answer to question (i).

Theorem 4.1. *The static planning problem (4.2)–(4.6) has a feasible solution with $\rho \leq 1$ if the network is stable under some service policy.*

We leave the proof of the theorem to Section 4.3. The next theorem, our main stability theorem, provides a complete answer to questions (i) and (ii). To state the theorem, we need to introduce an assumption on the network structure under which maximum pressure

policies are shown to be throughput optimal. For an allocation $a \in \mathcal{A}$, buffer i is said to be a constituent buffer of a if it can generate positive flow under a , i.e., $\sum_{j \in \mathcal{J}} a_j B_{ji} > 0$.

Assumption 4.1 (EAA assumption). For any buffer level vector $z \in \mathbb{R}_+^I$, there exists an extreme allocation $a^* \in \mathcal{E}$ that maximizes the network pressure $p(a, z)$, i.e., $p(a^*, z) = \max_{a \in \mathcal{E}} p(a, z)$, and that for each constituent buffer i of a^* , the buffer level z_i is positive.

The above assumption is called the extreme-allocation-available (EAA) assumption. It basically ensures that the maximum pressure allocation in (3.3) can be achieved by some feasible extreme allocation, i.e., $\max_{a \in \mathcal{A}} p(a, Z(t)) = \max_{a \in \mathcal{E}(t)} p(a, Z(t))$, when each non-empty buffer has sufficiently many jobs. The EAA assumption is satisfied for a wide range of familiar networks including strict Leontief networks introduced in Bramson and Williams [16] and networks of switches (see Section 4.7.2). Assumption 4.1 fails to hold for some networks. In Section 4.4, we will discuss the assumption in more detail.

Theorem 4.2. *Consider a stochastic processing network that satisfies Assumption 4.1. The network operating under a preemptive, processor-splitting maximum pressure policy is pathwise stable if the static planning problem (4.2)–(4.6) has a feasible solution with $\rho \leq 1$.*

The proof of Theorem 4.2 will be given in Section 4.2 with some of the supporting results proved in Section 4.3. The maximum pressure policy can be generalized in the following way. For each buffer i , let γ_i be a positive number and θ_i be a real number. Given parameters $\gamma = (\gamma_i)$ and $\theta = (\theta_i)$, define the new network pressure at time t under allocation a to be $p(a, \tilde{Z}(t))$, where $\tilde{Z}_i(t) = \gamma_i Z_i(t) - \theta_i$. The parameterized maximum pressure policies associated with parameters γ and θ can be defined through the total network pressure as before. For the parameterized maximum pressure policies, we have the following corollary.

Corollary 4.1. *Consider a stochastic processing network that satisfies Assumption 4.1. The network operating under a parameterized, preemptive, processor-splitting maximum pressure policy is pathwise stable if the static planning problem (4.2)–(4.6) has a feasible solution with $\rho \leq 1$.*

Since the proofs of Theorem 4.2 and Corollary 4.1 are identical, to keep notation simple,

we only consider the maximum pressure policies defined in Definition 3.1 except otherwise mentioned.

4.2 *Fluid Models and an Outline of the Proof of Theorem 4.2*

To prove Theorems 4.1 and 4.2, we adopt the standard fluid model approach [22]. In addition to introducing fluid models and their stability, this section outlines a proof of Theorem 4.2. The outline provides some key insights as to why a maximum pressure policy can stabilize a stochastic processing network.

The fluid model of a stochastic processing network is the deterministic, continuous analog of the stochastic processing network. It is defined by the following equations:

$$\bar{Z}(t) = \bar{Z}(0) + R\bar{T}(t), \quad (4.7)$$

$$\bar{Z}(t) \geq 0, \quad (4.8)$$

$$\sum_{j \in \mathcal{J}} A_{kj} (\bar{T}_j(t) - \bar{T}_j(s)) = t - s \text{ for each } 0 \leq s \leq t \text{ and each input processor } k, \quad (4.9)$$

$$\sum_{j \in \mathcal{J}} A_{kj} (\bar{T}_j(t) - \bar{T}_j(s)) \leq t - s \text{ for each } 0 \leq s \leq t \text{ and each processor } k, \quad (4.10)$$

$$\bar{T} \text{ is nondecreasing and } \bar{T}(0) = 0. \quad (4.11)$$

Equations (4.7)–(4.11) are analogous to stochastic processing network equations (2.5)–(2.9). They define the fluid model under any given service policy. Any quantity (\bar{Z}, \bar{T}) that satisfies (4.7)–(4.11) is a *fluid model solution* to the fluid model that operates under a general service policy. Following its stochastic processing network counterparts discussed in Section 2, each fluid model solution (\bar{Z}, \bar{T}) has the following interpretations: $\bar{Z}_j(t)$ the fluid level in buffer i at time t and $\bar{T}_j(t)$ the cumulative amount of activity j processing time in $[0, t]$.

For each fluid model solution (\bar{Z}, \bar{T}) , it follows from equations (4.9)–(4.10) that \bar{T} , and hence \bar{Z} , is Lipschitz continuous. Thus, the solution is absolutely continuous and has derivatives almost sure everywhere with respect to Lebesgue measure on $[0, \infty)$. A time $t > 0$ is said to be a regular point of the fluid model solution if the solution is differentiable

at time t . For a function $f : \mathbb{R}_+ \rightarrow \mathbb{R}^d$ for some positive integer d , we use $\dot{f}(t)$ to denote the derivative of f at time t when the derivative exists. From (4.9)-(4.10), one has $\dot{T}(t) \in \mathcal{A}$ at each regular time t . Thus, for a fluid model solution (\bar{Z}, \bar{T}) under a general service policy, the network pressure $R\dot{T}(t) \cdot \bar{Z}(t)$ under allocation $\dot{T}(t)$ when the fluid level is $\bar{Z}(t)$ is less than or equal to the maximum pressure $\max_{a \in \mathcal{A}} Ra \cdot \bar{Z}(t)$. Namely,

$$R\dot{T}(t) \cdot \bar{Z}(t) \leq \max_{a \in \mathcal{A}} Ra \cdot \bar{Z}(t) = \max_{a \in \mathcal{E}} Ra \cdot \bar{Z}(t). \quad (4.12)$$

Under a specific service policy, there are additional fluid model equations. Under a maximum pressure policy and the EAA assumption,

$$R\dot{T}(t) \cdot \bar{Z}(t) = \max_{a \in \mathcal{E}} Ra \cdot \bar{Z}(t) \quad (4.13)$$

for each regular time t . Thus, under a maximum pressure policy, the instantaneous activity allocation $\dot{T}(t)$ in the fluid model maximizes the network pressure at time t . Each fluid model equation will be justified through a fluid limit procedure. Three types of fluid limits are considered in this thesis. They will be introduced in Section 4.3, Section 5.5 and Appendix B, respectively. They all satisfy the fluid model equations (4.7)–(4.11) and (4.13). Any fluid model solution that satisfies fluid model equations (4.7)–(4.11) and (4.13) is called a *fluid model solution under the maximum pressure policy*.

We now give another interpretation of a maximum pressure policy. Let (\bar{Z}, \bar{T}) be a fluid model solution. Consider the following quadratic Lyapunov function:

$$f(t) = \sum_i \bar{Z}_i^2(t). \quad (4.14)$$

At a regular time t ,

$$\dot{f}(t) = 2\dot{\bar{Z}}(t) \cdot \bar{Z}(t) = -2R\dot{T}(t) \cdot \bar{Z}(t), \quad (4.15)$$

where, in the second equality, we have used the vector form of fluid model equation (4.7)

$$\dot{\bar{Z}}(t) = -R\bar{T}(t). \quad (4.16)$$

It follows from (4.12), (4.13) and (4.15) that the derivative of “system energy” $f(t)$ is minimized when $\dot{T}(t)$ is chosen as a fluid model solution under a maximum pressure policy.

Thus, a maximum pressure policy is *system greedy* in that the “system energy” decreases fastest (or increases slowest) at any regular time.

In addition to providing interpretations of a maximum pressure policy in the fluid model setting, the fluid model allows us to prove our main theorems. The following theorem provides a connection between the stability of a stochastic processing network and the weak stability of the corresponding fluid model, a notion we first define now.

Definition 4.2. A fluid model is said to be *weakly stable* if for every fluid model solution (\bar{Z}, \bar{T}) with $\bar{Z}(0) = 0$, $\bar{Z}(t) = 0$ for $t \geq 0$.

Theorem 4.3. *For a stochastic processing network, if the corresponding fluid model is weakly stable, it is pathwise stable.*

The proof of Theorem 4.3 will be presented in Section 4.3. In light of the theorem, the following theorem provides a complete proof of Theorem 4.2.

Theorem 4.4. *Assume that the linear program (4.2)-(4.6) has a feasible solution with $\rho \leq 1$ and the EAA assumption is satisfied. The fluid model under a maximum pressure policy is weakly stable.*

Proof. Let (\bar{Z}, \bar{T}) be a fluid model solution with $\bar{Z}(0) = 0$, and f be the quadratic Lyapunov function as defined in (4.14). Let x^* be a solution to the linear program (4.2)-(4.6) with $\rho \leq 1$. Obviously $x^* \in \mathcal{A}$ since $\rho \leq 1$. Moreover $Rx^* = 0$. It follows from (4.13) and (4.15) that

$$\begin{aligned} \dot{f}(t) &= -2R\dot{\bar{T}}(t) \cdot \bar{Z}(t) \\ &= -2 \max_{a \in \mathcal{A}} Ra \cdot \bar{Z}(t) \\ &\leq -2Rx^* \cdot \bar{Z}(t) \\ &= 0 \end{aligned}$$

for each regular time t . Since $f(0) = 0$, we have $f(t) = 0$ for $t \geq 0$, proving the fluid model is weakly stable. □

We end this section by stating a stronger version of Theorem 4.4. This result is of independent interest. We first make another minor assumption on the stochastic processing network.

Assumption 4.2. There exists an $x \geq 0$ such that $Rx > 0$.

Definition 4.3. A fluid model is said to be *stable* if there exists a constant $\delta > 0$ such that for every fluid model solution (\bar{Z}, \bar{T}) with $|\bar{Z}(0)| \leq 1$, $\bar{Z}(t) = 0$ for $t \geq \delta$.

Theorem 4.5. *For a stochastic processing network satisfying Assumptions 4.1 and 4.2, the corresponding fluid model is stable if the linear program (4.2)–(4.6) has a feasible solution with $\rho < 1$.*

Proof. Suppose that (\tilde{x}, ρ) is a feasible solution to (4.2)–(4.6). From Assumption 4.2, there exists an $\hat{x} \geq 0$ such $R\hat{x} > 0$. Since for each input activity j , $R_{ij} = -\mu_j B_{j0} P_{0i}^j \leq 0$, we can set $\hat{x}_j = 0$ for each input activity j so that $R\hat{x} > 0$ still holds. As a consequence, $\sum_j A_{kj} \hat{x}_j = 0$ for each input processor k . Clearly, \hat{x} can be scaled so that $\sum_j A_{kj} \hat{x}_j \leq (1 - \rho)$ for each service processor k . Let $x^* = \tilde{x} + \hat{x}$. One can check that $x^* \in \mathcal{A}$, and $Rx^* = R\tilde{x} + R\hat{x} = R\hat{x} \geq \delta e$, where $\delta = \min_i \sum_j R_{ij} \hat{x}_j > 0$. By (4.13),

$$R\dot{\bar{T}}(t) \cdot \bar{Z}(t) \geq Rx^* \cdot \bar{Z}(t) \geq \delta \sum_i \bar{Z}_i(t) \geq \delta \|\bar{Z}(t)\|,$$

where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^{\mathbf{I}}$. Therefore,

$$\dot{f}(t) = 2\dot{\bar{Z}}(t) \cdot \bar{Z}(t) = -2R\dot{\bar{T}}(t) \cdot \bar{Z}(t) \leq -2\delta \|\bar{Z}(t)\| = -2\delta \sqrt{f(t)}.$$

It follows that $\bar{Z}(t) = 0$ for $t \geq \|\bar{Z}(0)\|/\delta$. □

4.3 Fluid Limits

In this section, we introduce fluid limits that connect a stochastic processing network with the corresponding fluid model introduced in Section 4.2. As a consequence, we show that the stability of a fluid model implies the stability of the corresponding stochastic processing network (Theorem 4.3).

Recall that $\mathbb{X} = (Z, T)$ is the stochastic network process describing the stochastic processing network, where $Z_i(t)$ is the number of jobs at time t in buffer i , and $T_j(t)$ is the

cumulative activity j processing time in $[0, t]$. Clearly, \mathbb{X} depends on a realization of sample path ω . We use $\mathbb{X}(\cdot, \omega)$ to denote the trajectory of the network process along sample path ω .

For each $r > 0$ and ω , define the scaled process \mathbb{X}^r via

$$\bar{\mathbb{X}}^r(t, \omega) = r^{-1}\mathbb{X}(rt, \omega) \text{ for each } t \geq 0.$$

By strong law of large numbers, we have, with probability one,

$$\lim_{n \rightarrow \infty} \sum_{\ell=1}^n u_j(\ell)/n = 1 \text{ for each } j \in \mathcal{J}, \quad (4.17)$$

$$\lim_{\ell \rightarrow \infty} \Phi_i^j(\ell)/\ell = P_i^j \text{ for each } j \in \mathcal{J} \text{ and } i \in \mathcal{B}_j. \quad (4.18)$$

Definition 4.4. A function $\bar{\mathbb{X}} = (\bar{Z}, \bar{T})$ is said to be a *fluid limit* of the processing network if there exists a sequence $r \rightarrow \infty$ and a sample path ω satisfying (4.17)–(4.18) such that

$$\lim_{r \rightarrow \infty} \bar{\mathbb{X}}^r(\cdot, \omega) \rightarrow \bar{\mathbb{X}}(\cdot).$$

To see the existence of a fluid limit, note that for each activity j and each sample path ω , for all $r > 0$

$$\bar{T}_j^r(t, \omega) - \bar{T}_j^r(s, \omega) \leq t - s \text{ for } 0 \leq s \leq t.$$

Thus, the family of functions $\bar{T}_j^r(\cdot, \omega)$, $r > 0$, is equi-continuous, see for example Royden [59].

Thus, there is a subsequence $r_n \rightarrow \infty$ as $n \rightarrow \infty$ such that

$$\lim_{n \rightarrow \infty} \bar{T}_j^{r_n}(\cdot, \omega) \rightarrow \bar{T}_j(\cdot)$$

for some continuous function $\bar{T}_j(\cdot)$. By a standard argument, one can find a further subsequence, still denoted by $\{r_n\}$ for notational convenience, such that

$$\lim_{n \rightarrow \infty} \bar{T}_j^{r_n}(\cdot, \omega) = \bar{T}_j(\cdot) \quad (4.19)$$

for each activity j .

To show that $\bar{Z}^{r_n}(\cdot, \omega)$ converges, we are going to use flow balance equation (2.5). Let us first focus on the last term in the right side of (2.5). By (4.17), for the fixed sample path ω ,

$$\lim_{r \rightarrow \infty} S_j(rt, \omega)/r = \mu_j t \text{ for each } t \geq 0. \quad (4.20)$$

By (4.19) and (4.20), for each fixed t ,

$$\begin{aligned} \lim_{n \rightarrow \infty} \sum_j S_j(T_j(r_n t), \omega) / r_n &= \sum_j \lim_{n \rightarrow \infty} \frac{S_j(T_j(r_n t), \omega)}{T_j(r_n t)} \frac{T_j(r_n t)}{r_n} \\ &= \sum_{j \in \mathcal{J}} \mu_j \bar{T}_j(t). \end{aligned} \quad (4.21)$$

Since for each fixed n , $\sum_j S_j(T_j(r_n t), \omega) / r_n$ is a non-decreasing function of t , and the limit function $\sum_{j \in \mathcal{J}} \mu_j \bar{T}_j(t)$ is a continuous function of t , convergence in (4.21) actually holds uniformly on compact sets; see, for example, Lemma 4.1 of Dai [21].

Similarly, by using (4.18), (4.19) and (4.20), we have

$$\lim_{n \rightarrow \infty} \sum_j \sum_{i' \in \mathcal{B}_j} \Phi_{i'i}^j(S_j(T_j(r_n t))) / r_n = \sum_j \sum_{i' \in \mathcal{B}_j} \bar{T}_j(t) \mu_j P_{i'i}^j. \quad (4.22)$$

It follows from (2.5), (4.21), and (4.22) that $\bar{Z}^{r_n}(\cdot) \rightarrow \bar{Z}(\cdot)$ with $\bar{Z}(0) = 0$ and \bar{Z} satisfying (4.7). Clearly, $\bar{\mathbb{X}} = (\bar{Z}, \bar{T})$ also satisfies fluid model equations (4.8)-(4.11). Thus, the fluid limit $\bar{\mathbb{X}}$ is a fluid model solution to fluid model equations (4.7)-(4.11). We end this section by proving Theorems 4.1 and 4.3. Theorem 4.3 provides a connection between the stability of a stochastic processing network and the stability of the corresponding fluid model.

Proof of Theorem 4.1. Assume that the stochastic processing network is pathwise stable under some service policy. Fix a sample path that satisfies (4.1) and (4.17)-(4.18). Let (\bar{Z}, \bar{T}) be a fluid limit of (Z, T) along the sample path. Following the arguments in Section 4.3 such a limit exists and satisfies the fluid model equations (4.7)-(4.11). Since the stochastic processing network is stable, $\bar{Z}(t) = 0$ for $t \geq 0$. For each activity j , let $x_j = \bar{T}_j(1)$. It is easy to see that $x = (x_j)$ satisfies (4.2)-(4.6) with $\rho = 1$. \square

Proof of Theorem 4.3. Let ω be a sample path that satisfies (4.17) and (4.18). Let $\{r_n\}$ be a sequence such that $r_n \rightarrow \infty$ as $n \rightarrow \infty$. Consider the scaled sequence $\{\bar{\mathbb{X}}^{r_n}(t, \omega) = r_n^{-1} \mathbb{X}(r_n t, \omega), t \geq 0 : n \geq 1\}$. By the arguments in preceding three paragraphs, fluid limits exist. Namely, there exists a subsequence $\{r_{n'}\}$ such that

$$\bar{\mathbb{X}}^{r_{n'}}(\cdot, \omega) \rightarrow \bar{\mathbb{X}}(\cdot).$$

Since $\bar{\mathbb{X}} = (\bar{Z}, \bar{T})$ is also a fluid model solution with $\bar{Z}(0) = 0$, by the weak stability of the fluid model, $\bar{Z}(1) = 0$. Namely,

$$\lim_{n' \rightarrow \infty} \frac{Z(r_{n'}, \omega)}{r_{n'}} = 0. \quad (4.23)$$

Since $\{r_n\}$ is an arbitrary sequence with $r_n \rightarrow \infty$, (4.23) implies that

$$\lim_{t \rightarrow \infty} \frac{Z(t, \omega)}{t} = 0.$$

Thus, the stochastic processing network is pathwise stable. \square

4.3.1 Fluid limits under a maximum pressure policy

The main purpose of this section is to prove the following lemma, justifying that the fluid model under a maximum pressure policy is well defined by fluid model equations (4.7)-(4.11) and (4.13).

Lemma 4.1. *Consider a stochastic processing network operating under a processor-splitting, preemptive maximum pressure policy. Assume the network satisfies Assumption 4.1. Each fluid limit satisfies fluid model equation (4.13).*

The proof of the lemma will be presented at the end of this section. For that, let $T^a(t)$ be the cumulative amount of time that allocation a has been employed in $[0, t]$. Under a maximum pressure policy, only extreme allocations are used. Thus,

$$T_j(t) = \sum_{a \in \mathcal{E}} a_j T^a(t) \quad \text{for each } t \geq 0 \text{ and } j \in \mathcal{J}. \quad (4.24)$$

Clearly,

$$T^a(\cdot) \text{ is nondecreasing for each allocation } a \in \mathcal{E}, \quad (4.25)$$

$$\sum_{a \in \mathcal{E}} T^a(t) = t \text{ for each } t \geq 0. \quad (4.26)$$

Since each extreme allocation is in \mathcal{A} , one can check that (4.24)–(4.26) imply (2.7) and (2.8).

Under a maximum pressure policy, we modify the definition of the stochastic processing network process via

$$\mathbb{X}(t) = (Z(t), T^a(t), T_j(t) : a \in \mathcal{E}, j \in \mathcal{J}).$$

Each \mathbb{X} satisfies the stochastic processing network equations (2.5), (2.6), and (4.24)-(4.26).

The fluid limits of \mathbb{X} are defined analogously as in Section 4.3.

Lemma 4.2. *Assume that the EAA assumption holds. Each fluid limit $\bar{\mathbb{X}} = (\bar{Z}, \bar{T}^a, \bar{T}_j : a \in \mathcal{E}, j \in \mathcal{J})$ under a preemptive, processor-splitting maximum pressure policy satisfies fluid model equations (4.7)-(4.11), and the following equations*

$$\bar{T}_j(t) = \sum_{a \in \mathcal{E}} a_j \bar{T}^a(t) \text{ for each } t \geq 0 \text{ and } j \in \mathcal{J}, \quad (4.27)$$

$$\bar{T}^a(\cdot) \text{ is nondecreasing for each allocation } a \in \mathcal{E}, \quad (4.28)$$

$$\sum_{a \in \mathcal{E}} \bar{T}^a(t) = t \text{ for each } t \geq 0, \quad (4.29)$$

$$\dot{\bar{T}}^a(t) = 0, \text{ if } p(a, \bar{Z}(t)) < \max_{a' \in \mathcal{E}} p(a', \bar{Z}(t)), \quad (4.30)$$

Proof. Let $\bar{\mathbb{X}}$ be a fluid limit. Clearly, it satisfies (4.27)-(4.29). It remains to prove that $\bar{\mathbb{X}}$ satisfies (4.30).

Recall that, $p(a, \bar{Z}(t)) = \bar{Z}(t) \cdot Ra$ is the network pressure under allocation a when the fluid level is $\bar{Z}(t)$. Suppose that $a \in \mathcal{E}$ and $p(a, \bar{Z}(t)) < \max_{a' \in \mathcal{E}} p(a', \bar{Z}(t))$. From Assumption 4.1, we can choose an $a^* \in \mathcal{E}$ such that

$$p(a^*, \bar{Z}(t)) = \max_{a' \in \mathcal{E}} p(a', \bar{Z}(t))$$

and the fluid level $\bar{Z}_i(t) > 0$ for each constituent buffer i of a^* . Denote $\mathcal{I}(a^*)$ the set of constituent buffers. Namely,

$$\mathcal{I}(a^*) = \left\{ i : \sum_j a_j^* B_{ji} > 0 \right\}.$$

Then $\bar{Z}_i(t) > 0$ for all $i \in \mathcal{I}(a^*)$. Since $p(a, \bar{Z}(t)) < p(a^*, \bar{Z}(t))$ and $\min_{i \in \mathcal{I}(a^*)} \bar{Z}_i(t) > 0$, by the continuity of $\bar{\mathbb{X}}(\cdot)$, there exist $\epsilon > 0$ and $\delta > 0$ such that for each $\tau \in [t - \epsilon, t + \epsilon]$ and $i \in \mathcal{I}(a^*)$,

$$p(a, \bar{Z}(\tau)) + \delta \leq p(a^*, \bar{Z}(\tau)) \quad \text{and} \quad \bar{Z}_i(\tau) \geq \delta.$$

Thus, when n is sufficiently large, $p(a, \bar{Z}(n\tau)) + n\delta/2 \leq p(a^*, \bar{Z}(n\tau))$ and $Z_i(n\tau) \geq n\delta/2$ for each $i \in \mathcal{I}(a^*)$ and each $\tau \in [t - \epsilon, t + \epsilon]$. Choosing $n > 2\mathbf{J}/\delta$, then for each $\tau \in$

$[n(t - \epsilon), n(t + \epsilon)]$ we have

$$p(a, Z(\tau)) < p(a^*, Z(\tau)), \quad (4.31)$$

$$Z_i(\tau) \geq \mathbf{J} \quad \text{for each } i \in \mathcal{I}(a^*). \quad (4.32)$$

Condition (4.32) implies that a^* is a feasible allocation at any time $\tau \in [n(t - \epsilon), n(t + \epsilon)]$, i.e., $a^* \in \mathcal{E}(\tau)$. Following (4.31) and the definition of a (preemptive-resume) maximum pressure policy, the allocation a will not be employed during time interval $[n(t - \epsilon), n(t + \epsilon)]$. Therefore,

$$T^a(n(t + \epsilon)) - T^a(n(t - \epsilon)) = 0, \quad (4.33)$$

which implies $\bar{T}^a(t + \epsilon) - \bar{T}^a(t - \epsilon) = 0$, and hence $\dot{\bar{T}}^a(t) = 0$.

□

Proof of Lemma 4.1. Let $\bar{\mathbf{X}}$ be a fluid limit. We would like to prove that (\bar{Z}, \bar{T}) satisfies fluid model equation (4.13). By Lemma 4.2 and the fact that $\sum_{a \in \mathcal{E}} \dot{\bar{T}}^a(t) = 1$,

$$\sum_{a \in \mathcal{E}} \dot{\bar{T}}^a(t) p(a, \bar{Z}(t)) \geq p(a', \bar{Z}(t)) \quad \text{for all } a' \in \mathcal{E}.$$

Now,

$$\begin{aligned} R\dot{\bar{T}}(t) \cdot \bar{Z}(t) &= \sum_{i \in \mathcal{I}} \bar{Z}_i(t) \sum_{j \in \mathcal{J}} R_{ij} \dot{\bar{T}}_j(t) = \sum_{i \in \mathcal{I}} \bar{Z}_i(t) \sum_{j \in \mathcal{J}} R_{ij} \sum_{a' \in \mathcal{E}} a'_j \bar{T}^{a'}(t) \\ &= \sum_{a' \in \mathcal{E}} \dot{\bar{T}}^{a'}(t) p(a', \bar{Z}(t)) \\ &\geq p(a, \bar{Z}(t)) = Ra \cdot \bar{Z}(t). \end{aligned}$$

The preceding inequality together with (4.12) proves (4.13).

□

4.4 EAA Assumption Revisited

In this section, we introduce *strict Leontief networks* and verify that the extreme-allocation-available (EAA) assumption, Assumption 4.1, is always satisfied for such networks. We then present a network example for which the EAA assumption fails to satisfy.

4.4.1 Strict Leontief networks

A stochastic processing network is said to be *strict Leontief* if each service activity is associated with exactly one buffer, i.e., \mathcal{B}_j contains exactly one buffer for each activity $j \in \mathcal{J}$. (Recall that each input activity j is assumed to be associated with buffer 0 only, i.e., $\mathcal{B}_j = \{0\}$.)

Theorem 4.6. *Assumption 4.1 is satisfied for strict Leontief networks.*

Proof. For strict Leontief networks, each row j of matrix B has exactly one entry equal to 1. Denote the corresponding buffer as $i(j)$. Then $B_{ji} = 0$ and $R_{ij} \leq 0$ for each $i \in \mathcal{I}$ and $j \in \mathcal{J}$ such that $i \neq i(j)$. For any vector $z \in \mathbb{R}_+^{\mathcal{I}}$, we define \mathcal{J}_0 as the set of service activities j with $z_{i(j)} = 0$. It is sufficient to show that there exists an allocation $a^* \in \arg\max_{a \in \mathcal{A}} z'Ra$ with $a_j^* = 0$ for all $j \in \mathcal{J}_0$. Let \hat{a} be any extreme allocation such that $z'R\hat{a} = \max z'Ra$. Define \tilde{a} via

$$\tilde{a}_j = \begin{cases} 0, & j \in \mathcal{J}_0, \\ \hat{a}_j, & j \notin \mathcal{J}_0. \end{cases}$$

Obviously, $\tilde{a} \in \mathcal{A}$. Note that

$$z'R\hat{a} = \sum_i \sum_j z_i R_{ij} \hat{a}_j = \sum_i \sum_{j \in \mathcal{J}_0} z_i R_{ij} \hat{a}_j + \sum_i \sum_{j \in \mathcal{J} \setminus \mathcal{J}_0} z_i R_{ij} \hat{a}_j = \sum_i \sum_{j \in \mathcal{J}_0} z_i R_{ij} \hat{a}_j + z'R\tilde{a}.$$

Since

$$\sum_i \sum_{j \in \mathcal{J}_0} z_i R_{ij} \hat{a}_j = \sum_{j \in \mathcal{J}_0} \sum_{i \neq i(j)} z_i R_{ij} \hat{a}_j \leq 0,$$

we have $z'R\tilde{a} \geq z'R\hat{a}$. Because $z'R\hat{a} = \max_{a \in \mathcal{A}} z'Ra$, we have $z'R\tilde{a} = \max_{a \in \mathcal{A}} z'Ra$. If \tilde{a} is an extreme allocation, then the proof is complete. Otherwise, it is a linear combination of some extreme allocations. Choose any one of these allocations as a^* . Then $z'Ra^* = \max_{a \in \mathcal{A}} z'Ra$, and $a_j^* = 0$ for all $j \in \mathcal{J}_0$. \square

4.4.2 A network example not satisfying the EAA assumption

In this section, we present a network example for which the EAA assumption is not satisfied. The network has a feasible solution with $\rho < 1$ to the static planning problem (4.2)–(4.6), yet it is not stable under any maximum pressure policy. The network is shown to be stable under some allocation policy.

Consider the network depicted in Figure 4.1. The network has 5 internal buffers, in addition to the external buffer 0. Buffers are represented by open rectangular boxes. There are 6 processors represented by circles, and 8 activities labeled on the lines connecting circles to buffers. All processors, except processor 6, are input processors. For $\ell = 1, \dots, 5$, input processor ℓ works on activity ℓ to generate the arrivals to buffer ℓ with rate μ_ℓ . Each time, the service processor, processor 6, can employ one activity from activities 6, 7, and 8. By employing activity 6, the service processor processes jobs from all the five buffers simultaneously with rate μ_6 . By employing activity 7 (or 8), the service processor works simultaneously on buffers 1 and 2 (or 4 and 5) with rate μ_7 (or μ_8). It is clear that all input activities must always be employed. However each time only one service activity can be employed. Therefore, there are a total of 4 extreme allocations. They are $a^1 = (1, 1, 1, 1, 1, 0, 0)'$, $a^2 = (1, 1, 1, 1, 1, 0, 1, 0)'$, $a^3 = (1, 1, 1, 1, 1, 0, 0, 1)'$, and $a^4 = (1, 1, 1, 1, 1, 0, 0, 0)'$.

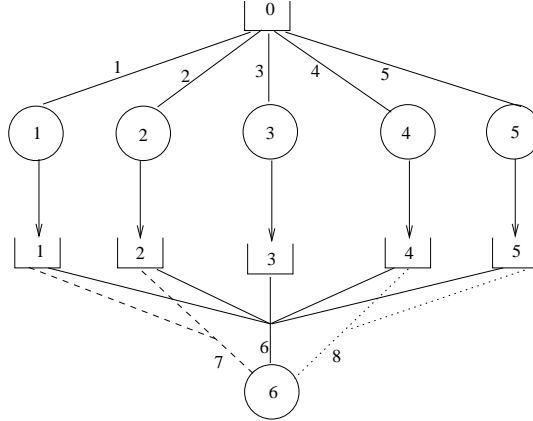


Figure 4.1: An example that does not satisfy the EAA assumption

We assume all the processing times are deterministic. Their values are specified as follows. Denote $\eta_j(\ell)$ to be the processing time of the ℓ th activity j . For activities $j = 1, 2$ and 3 , $\eta_j(\ell) = 2$, $\ell \geq 1$. For activities $j = 4$ and 5 , $\eta_j(1) = 1$, and $\eta_j(\ell) = 2$ for $\ell \geq 2$. For activities $j = 6, 7$ and 8 , $\eta_j(\ell) = 1$, $\ell \geq 1$. Therefore,

$$\mu_j = \begin{cases} 0.5, & j = 1, \dots, 5, \\ 1, & j = 6, 7, \text{ and } 8. \end{cases}$$

The input-output matrix R in (3.1) can be written as

$$R = \begin{pmatrix} -0.5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & -0.5 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & -0.5 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.5 & 1 & 0 & 1 \end{pmatrix}$$

It can be easily checked that $x = (1, 1, 1, 1, 1, 1/2, 0, 0)'$ and $\rho = 1/2$ is a feasible solution to the static planning problem (4.2)–(4.6).

The system is unstable under any maximum pressure policy. In the following, we show that for any maximum pressure policy with given parameters (γ, θ) , the queue size of buffer 3 can grow without bound under a certain initial condition. Without loss of generality, we assume that $\gamma_i = 1$ for each buffer i . Now consider the system with initial buffer sizes $Z_1(0) = \theta_1 + \theta_2$, $Z_5(0) = \theta_4 + \theta_5$, and $Z_i(0) = 0$ for $i = 2, 3$, and 4. At time $t < 1$, buffers 2, 3 and 4 are empty. Thus, none of the service activities can be employed, and the service processor is idle. At time $t = 1$, buffers 4 and 5 each have an arrival. Thus, $Z_4(t) = 1$ and $Z_5(t) = \theta_4 + \theta_5 + 1 \geq 1$, whereas buffers 2 and 3 remain empty. Therefore, allocations a^1 and a^2 are not feasible at time $t = 1$. Because buffers 4 and 5 have jobs at time $t = 1$, allocation a^3 is feasible. Furthermore, one can check that the network pressure $p(a^3, Z(t))$ under allocation a^3 is strictly larger than the network pressure $p(a^4, Z(t))$ under allocation a^4 . Under the maximum pressure policy, allocation a^3 will be employed at time $t = 1$. At time $t = 2$, buffers 1, 2 and 3 each have an arrival, and buffers 4 and 5 each have a departure. Thus, $Z_1(t) = \theta_1 + \theta_2 + 1 \geq 1$, $Z_2(t) = 1$, $Z_3(t) = 1$, and $Z_4(t) = 0$. Because $Z_4(t) = 0$, allocations a^1 and a^3 are not feasible. It is easy to verify that a^2 will be employed under the maximum pressure policy at time $t = 2$. At time $t = 3$, $Z_2(t) = 0$, $Z_3(t) = 1$, $Z_4(t) = 1$ and $Z_5(t) = \theta_4 + \theta_5 + 1 \geq 1$. One can argue similar to the reasoning at time $t = 1$ that allocation a^3 will be employed at time $t = 3$. At time $t = 4$, $Z_1(t) = \theta_1 + \theta_2 + 1 \geq 1$, $Z_2(t) = 1$, $Z_3(t) = 2$ and $Z_4(t) = 0$. One can argue similar to the reasoning at time $t = 2$ that allocation a^2 will be employed at time $t = 4$. Continuing this process, one realizes that at any time either buffer 2 or 4 will be empty. Therefore, activity 6 will always be infeasible,

and thus allocation a^1 is never feasible. Hence jobs in buffer 3 will never be processed. The system is unstable.

So far, we have verified that our network example has a feasible solution to (4.2)–(4.6). Yet none of the maximum pressure policies stabilize the network. These facts do not contradict Theorem 4.2 because the network does not satisfy the EAA assumption, a claim we now verify. Letting

$$z_j = \begin{cases} 0, & j \neq 3, \\ 1, & j = 3. \end{cases}$$

It is easy to see that $a^1 = \arg \max_{a \in \mathcal{E}} z \cdot Ra$ and a^1 is the only one that achieves the maximum. However, $a_6^1 = 1$ and $B_{61} = 1$, whereas $z_1 = 0$. This violates the EAA assumption.

A closer examination of the instability analysis of the maximum pressure policies reveals that under a maximum pressure policy, allocations a^2 and a^3 are so “greedily” employed that activity 6 never has a chance to be employed. If the service processor takes activity 6 when all five buffers are non-empty and idles otherwise, the system would be stable. To see this, we first assume that the system is initially empty. It is easy to see that $Z_4(t) = Z_5(t) = 1$ for all $t \geq 1$, and $Z_1(t) = Z_2(t) = Z_3(t) = 0$ for $t \in [2n + 1, 2n + 2)$, and $Z_1(t) = Z_2(t) = Z_3(t) = 1$ for $t \in [2n + 2, 2n + 3)$. Thus, (4.1) holds. For more general initial conditions, one can check that $Z_i(t) \leq Z_i(0) + 1$ for $i = 1, \dots, 5$. Thus, (4.1) holds in general, proving the stability of the network.

4.5 *Non-processor-splitting Service Policies*

In many applications including manufacturing systems, processor-splitting is not allowed. In this case, each allocation a has components that are either 1 or 0, i.e., $a \in \mathbb{Z}_+^{\mathbf{J}}$, where \mathbb{Z}_+ denotes the set of nonnegative integers. Since there are fewer possible allocations to choose from at each decision point, the stability results developed earlier may not hold anymore. In this section, we first provide an example for which, unlike Theorem 4.2, Harrison’s static planning problem (4.2)–(4.6) does not determine the stability region of a stochastic processing network when processor-splitting is not allowed. We then establish a

theorem that is analogous to Theorem 4.2 when processor-splitting is not allowed. Finally, we introduce a class of networks, called reversed Leontief networks, for which the static planning problem (4.2)–(4.6) still determines the stability regions even when processor-splitting is not allowed. Like previous sections, preemption is assumed throughout this section.

Consider the following example. As depicted in Figure 4.2, the network has 4 buffers (including buffer 0 representing the outside world), 6 processors, and 6 activities. Buffers are represented by open rectangular boxes, processors are represented by circles, and activities are labeled on lines connecting buffers with processors. Processors 1, 2, and 3 are input processors, and processors 4, 5, and 6 are service processors. Activities 1, 2, and 3 are input activities. Each input activity requires one input processor, taking input from buffer 0 and producing output to the corresponding buffer as indicated in the figure. Activities 4, 5 and 6 are service activities. Activity 4 requires processors 4 and 5 simultaneously to process jobs from buffer 1. Activity 5 requires processors 5 and 6 simultaneously to process jobs from buffer 2. Activity 6 requires processors 4 and 6 simultaneously to process jobs from buffer 3. The processing times of each activity are assumed to be deterministic. For each input activity i , the processing rate μ_i is assumed to be 0.4, $i = 1, 2, 3$. For each service activity i , the processing rate μ_i is assumed to be 1.0, $i = 4, 5, 6$. It follows from the definitions of input-output matrix R in (3.1) and the resource consumption matrix A that

$$R = \begin{pmatrix} -0.4 & 0 & 0 & 1 & 0 & 0 \\ 0 & -0.4 & 0 & 0 & 1 & 0 \\ 0 & 0 & -0.4 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

It is easy to check that $x = (1, 1, 1, 0.4, 0.4, 0.4)'$ and $\rho = 0.8$ is the optimal solution to the static planning problem (4.2)–(4.6). However, when processor-splitting is not allowed, we know that at any given time only one service activity can be active. Thus, the total maximum departure rate from the system is 1. But the total rate into buffers 1, 2 and 3 is

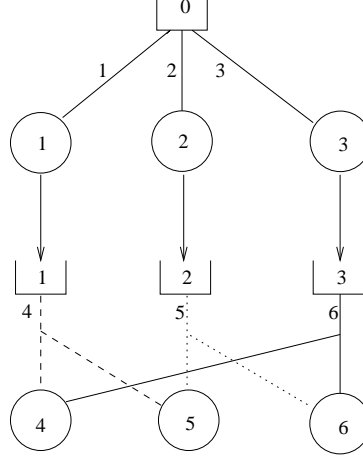


Figure 4.2: A processing network for which processor sharing is important

1.2, exceeding the total departure rate. Hence the system is unstable.

Note that the popular round-robin mechanism as in Andradottir et al [1] is a temporal way to share a processor's capacity. The above example can not be stabilized by any temporal sharing of the processors. However, it is stable if each service processor is shared at 50% level among its activities.

To develop an analogous stability theory in the non-processor-splitting case, we first change the allocation space \mathcal{A} to \mathcal{N} , where \mathcal{N} is the set of allocations $a \in \mathbb{Z}_+^{\mathbf{J}}$ that satisfy (2.2) and (2.3). Note that each allocation in \mathcal{N} is an extreme one. Using \mathcal{N} to replace \mathcal{E} , one can define a maximum pressure policy exactly as in Section 3. In the example described earlier in this section, $\mathcal{N} \neq \mathcal{E}$.

Consider the following static allocation problem (SAP): finding $\pi = (\pi_a)$ satisfying

$$Rx = 0, \tag{4.34}$$

$$x = \sum_{a \in \mathcal{N}} a \pi_a, \tag{4.35}$$

$$\sum_{a \in \mathcal{N}} \pi_a = 1, \tag{4.36}$$

$$\pi_a \geq 0 \text{ for each } a \in \mathcal{N}. \tag{4.37}$$

Here π_a is interpreted as the long run fraction of time that allocation a is employed, and x retains the same interpretation as in the static planning problem (4.3)–(4.6).

Theorem 4.7. *The static allocation problem (4.34)–(4.37) has a feasible solution if the stochastic processing network is pathwise stable under some non-processor-splitting service policy.*

Proof. Since the proof is analogous to the proof of Theorem 4.1, we present an outline, highlighting the differences between the two proofs. As in Section 4.3.1, for each allocation $a \in \mathcal{N}$, let $T^a(t)$ be the cumulative amount of time that allocation a has been employed in $[0, t]$. Since non-processor-splitting is assumed, equations (4.24)–(4.26) in Section 4.3.1 hold with \mathcal{N} replacing \mathcal{E} . Again as in Section 4.3.1, under a non-processor-splitting service policy, we modify the definition of the stochastic processing network process via

$$\mathbb{X}(t) = (Z(t), T^a(t), T_j(t) : a \in \mathcal{N}, j \in \mathcal{J}).$$

Each \mathbb{X} satisfies the stochastic processing network equations (2.5), (2.6), and (4.24)–(4.26). The fluid limits of \mathbb{X} are defined analogously as in Section 4.3. Assume that the stochastic processing network is pathwise stable under some non-processor-splitting service policy. Fix a sample path that satisfies (4.1) and (4.17)–(4.18). Let $(\bar{Z}, \bar{T}^a, \bar{T}_j : a \in \mathcal{N}, j \in \mathcal{J})$ be a fluid limit of \mathbb{X} along the sample path. Similar to the arguments in Section 4.3 such a limit exists and satisfies the fluid model equations (4.7)–(4.11), and (4.27)–(4.29) with \mathcal{N} replacing \mathcal{E} . Since the stochastic processing network is stable, $\bar{Z}(t) = 0$ for $t \geq 0$. For each allocation $a \in \mathcal{N}$, let $\pi_a = \bar{T}^a(1)$. It is easy to see that $\pi = (\pi_a)$ is a feasible solution satisfies (4.34)–(4.37). \square

Theorem 4.7 says that the feasibility of the static allocation problem (4.34)–(4.37) is necessary for the network to be pathwise stable under any non-processor-splitting policy. The following theorem asserts that if the static allocation problem (4.34)–(4.37) is feasible, the network is pathwise stable under a non-processor-splitting maximum pressure policy. Thus, non-processor-splitting maximum pressure policies are throughput optimal among non-processor-splitting policies.

Theorem 4.8. *Consider a stochastic processing network that satisfies Assumption 4.1 with \mathcal{E} replaced by \mathcal{N} . If the static allocation problem (4.34)–(4.37) has a feasible solution,*

the network operating under a non-processor-splitting maximum pressure policy is pathwise stable.

Proof. First under Assumption 4.1 with \mathcal{E} being replaced by \mathcal{N} , the fluid model operating under a non-processor-splitting maximum pressure is defined by equations (4.7)–(4.11) and

$$R\dot{T}(t) \cdot \bar{Z}(t) = \max_{a \in \mathcal{N}} Ra \cdot \bar{Z}(t). \quad (4.38)$$

The last equation replaces the fluid model equation (4.13) for the fluid model operating under a processor-splitting maximum pressure policy. With \mathcal{N} replacing \mathcal{A} , Lemmas 4.1 and 4.2 still hold. They provide a justification of fluid model equations (4.7)–(4.11) and (4.38), via fluid limits introduced in Section 4.3.

Let $\bar{\mathbb{X}} = (\bar{Z}, \bar{T})$ be a fluid model solution with $\bar{Z}(0) = 0$. Consider the quadratic Lyapunov function $f(t)$ defined in (4.14). It follows from (4.15) and (4.38) that, for any regular point t ,

$$\begin{aligned} \dot{f}(t) &= -2R\dot{T}(t) \cdot \bar{Z}(t) \\ &= -2 \max_{a \in \mathcal{N}} Ra \cdot \bar{Z}(t) \\ &\leq -2 \left(R \sum_{a \in \mathcal{N}} a\pi_a \right) \cdot \bar{Z}(t) \end{aligned}$$

for any distribution $\pi = (\pi_a)_{a \in \mathcal{N}}$ with $\pi_a \geq 0$ and $\sum_{a \in \mathcal{N}} \pi_a = 1$. Let π be a feasible solution to (4.34)–(4.37). Then,

$$R \sum_{a \in \mathcal{N}} a\pi_a = 0$$

and $\sum_{a \in \mathcal{N}} \pi_a = 1$. Thus, $\dot{f}(t) \leq 0$. Hence $\bar{Z}(t) = 0$ for $t \geq 0$, proving the weak stability of the fluid model. By Theorem 4.3, the stochastic processing network is pathwise stable. \square

When $\mathcal{N} = \mathcal{E}$, the maximum pressure policies used in Chapter 3 are actually non-processor-splitting. Therefore, Theorem 4.2 still holds in this case.

Definition 4.5. A stochastic processing network is said to be *reversed Leontief* if each activity requires exactly one processor.

Lemma 4.3. *For a reversed Leontief network, every extreme allocation is an integer allocation, i.e., $\mathcal{E} = \mathcal{N}$.*

Proof. For each processor k , let $\mathcal{J}(k)$ be the set of possible activities that the processor can take. We make the convention that when a processor is idle, it takes on activity 0. (Note that idling is not an activity as defined in Section 2.) Thus,

$$\mathcal{J}(k) = \begin{cases} \{j : A_{kj} = 1\}, & \text{for input processor } k, \\ \{0\} \cup \{j : A_{kj} = 1\}, & \text{for service processor } k. \end{cases}$$

We prove the lemma by contradiction. Suppose that there exists an allocation $a \in \mathcal{E}$ such that $0 < a_{\tilde{j}} < 1$ for some activity $\tilde{j} \in \mathcal{J}$. Let \tilde{k} be the processor processing activity \tilde{j} . For each $j \in \mathcal{J}(\tilde{k})$, we define a new allocation b^j by modifying the allocation a in the following way. For activities $j' \notin \mathcal{J}(\tilde{k})$, we keep the activity level $a_{j'}$; processor \tilde{k} employs activity j at 100% level. Clearly, b^j is a feasible allocation for each $j \in \mathcal{J}(\tilde{k})$. It follows that $a = \sum_{j \in \mathcal{J}(\tilde{k})} a_j b^j$, where, for a service processor \tilde{k} , we set $a_0 = 1 - \sum_{j \in \mathcal{J}(\tilde{k}), j \neq 0} a_j$. Since $\sum_{j \in \mathcal{J}(\tilde{k})} a_j = 1$, $\tilde{j} \in \mathcal{J}(\tilde{k})$, and $a_{\tilde{j}} < 1$ by assumption, a is a proper linear combination of feasible allocations. Thus, a is not an extreme allocation, contradicting the assumption that a is an extreme allocation. Therefore, any extreme allocation must be an integer allocation. On the other hand, any feasible integer allocation must be extreme. Hence $\mathcal{E} = \mathcal{N}$. \square

From the lemma, the maximum pressure policies in a reversed Leontief network are always throughput optimal whether processor-splitting is allowed or not. The resulting allocations are always non-processor-splitting.

4.6 Non-preemptive Service Policies

In the definition of maximum pressure policies, we have implicitly assumed that preemption of activities is allowed. These policies are defined through extreme allocations. When preemption is not allowed, at any given time the feasible set of allocations is reduced. It is possible that this feasible set does not contain any extreme allocations, yielding the current definition of the policy invalid in the non-preemption case. When preemption is not allowed, even when a maximum pressure policy can be defined, the example to be

presented below shows that Theorem 4.2 does not hold in general. In other words, even if the static planning problem (4.2)–(4.6) has a feasible solution, the network is not stable under a non-preemption version of a maximum pressure policy. In this section, we will first present an example. We then prove that when a stochastic processing network has some special structure, a non-preemptive maximum pressure service policy is well defined and is throughput optimal.

Consider the example depicted in Figure 4.3 with 2 service processors processing jobs in buffers 1 and 2. Jobs arrive at buffer 1 at time $0.5 + 2n$, $n = 0, 1, 2, \dots$, and arrive at buffer 2 at time $1 + 1.5n$, $n = 0, 1, 2, \dots$. The network is assumed to be empty initially. There are three service activities: activity 1 requires service processor 1 and processes jobs in buffer 1 with a deterministic processing requirement of 1 unit of time; activity 2 requires service processor 2 and processes jobs in buffer 2 with a deterministic processing requirement of 2 units of time; and activity 3 requires both servers 1 and 2 and processes jobs in buffer 2 with a deterministic processing requirement of 1 unit of time. (Input activities and buffer 0 representing the outside world are not drawn in the figure.)

One can easily verify that the static planning problem (4.2)–(4.6) has a feasible solution with $\rho = 11/12$. Since each activity processes jobs from in a single buffer, the network is strictly Leontief. By Theorem 4.6, the EAA assumption is satisfied. Hence any preemptive, processor-splitting maximum pressure policy is pathwise stable for the network. We now show that a non-preemptive maximum pressure policy is not pathwise stable. At time $t = 0.5$, there is an arrival at buffer 1 and activity 1 becomes active because the allocation $(1, 0, 0)$ is the maximum pressure allocation. At time $t = 1$, there is an arrival at buffer

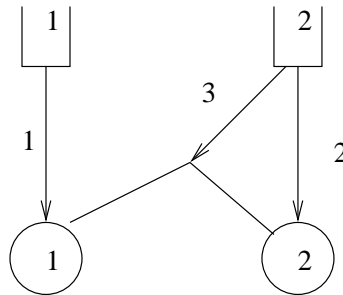


Figure 4.3: Non-preemption could make unstable

2. Because processor 1 has been assigned to activity 1 which cannot be interrupted, only allocations $(1, 1, 0)$ and $(1, 0, 0)$ are feasible at $t = 1$. It can be verified that allocation $(1, 1, 0)$ is the maximum pressure allocation at this time. Therefore, activity 2 becomes active at time $t = 1$. At time $t = 1.5$, processor 1 completes the activity 1 processing (of the job in buffer 1). Because processor 2 is still active processing activity 2, the only feasible allocation during time interval $[1.5, 2.5]$ is $(0, 1, 0)$. At time $t = 2.5$, activity 1 will be active again. It is easy to verify that activity 1 will be active during time interval $[0.5 + 2n, 1.5 + 2n]$ and activity 2 completes a processing at time $1 + 2n$ at which the processor 1 is tied with an activity 1 processing. Thus, processors 1 and 2 have no chance to engage activity 3. Therefore, under the non-preemptive maximum pressure policy, the departure rate of buffer 2 is $1/2$, which is less than the arrival rate $2/3$. Hence, the network is not pathwise stable.

Recall that a stochastic processing network is called a reversed Leontief network if each activity needs exactly one processor to be active. By Lemma 4.3, each maximum pressure policy is necessarily non-processor-splitting. We now show that, in a reversed Leontief network, the non-preemptive version of a maximum pressure policy is well defined. Furthermore, the network operating under such a policy is throughput optimal.

The following lemma shows that the non-preemption maximum pressure policies are well defined for reversed Leontief networks.

Lemma 4.4. *For reversed Leontief networks, the non-preemption maximum pressure policies are well defined. That is, $\mathcal{E}(t)$ is non-empty for each time $t \geq 0$.*

Proof. It is easy to see that there exists a feasible extreme allocation initially. For instance, one can let each input processor choose any associated activity and let all service processors idle. Now suppose that a non-preemptive maximum pressure policy is well defined prior to time t . We consider two cases, depending on if there is a service activity completion at time t . (1) Assume there is no service activity completion at time t . For this case, the previous allocation is still feasible and also extreme. Thus, $\mathcal{E}(t)$ is not empty. (2) Assume that there is a service activity completion at time t . Let a be the allocation immediately

preceding t . By assumption, a is extreme. We now define a new allocation \tilde{a} . We let $\tilde{a}_j = a_j$ for all activities j except the service activities that have completions at time t . For these service activities j , we let $\tilde{a}_j = 0$. Clearly, \tilde{a} is still extreme. Since each service activity is associated with exactly one processor, allocation \tilde{a} is a feasible allocation. Thus, $\mathcal{E}(t)$ is not empty. \square

In the rest of this section, we will prove a theorem that non-preemption maximum pressure policies are throughput optimal in reversed Leontief networks. To state the theorem, we need a slightly stronger assumption on the service times than the one in (4.17).

Assumption 4.3. For each activity $j \in \mathcal{J}$, there exist $\epsilon_j > 0$ and $m'_j > 0$ such that, with probability one,

$$\lim_{n \rightarrow \infty} n^{-1} \sum_{\ell=1}^n \eta_j^{1+\epsilon_j}(\ell) = m'_j, \quad (4.39)$$

where $\eta_j(\ell) = u_j(\ell)/\mu_j$ is the processing time of the ℓ th activity j .

Theorem 4.9. *Assume Assumption 4.1 and Assumption 4.3 are satisfied, then the reversed Leontief networks operating under non-preemptive, non-processor-splitting maximum pressure policies are pathwise stable if the static planning problem (4.2)–(4.6) has a feasible solution (x, ρ) with $\rho \leq 1$.*

Remark. In some networks, preempting service activities is allowed, but preempting input activities is not. In this case, if the input processors and input activities satisfy the reversed Leontief property, the stability result analogous to Theorem 4.9 still holds.

To prove Theorem 4.9, in light of the proof of Theorem 4.2 and Lemma 4.5, it is sufficient to prove that the fluid model for a reserved Leontief network operating under a non-preemption maximum pressure policy is still defined by (4.7)–(4.11) and (4.13). We leave the proof to the end of this Section. The proof needs the following lemma, which may be of independent interest.

To present the lemma, recall that in a reversed Leontief network, by Lemma 4.5, each extreme allocation a is an integer allocation. Thus, each processor k is employed by at most one activity. We use $j_k(a)$ to denote the activity that processor k is working on under

allocation a . We set $j_k(a) = 0$ if processor k is idle under allocation a . Recall that $\mathcal{J}(k)$ is the set of possible activities that processor k can take, including the idle activity 0.

For an activity $j \in \mathcal{J}$, define activity j pressure when the buffer level is $Z(t)$ to be

$$p(j, Z(t)) = \sum_{i \in \mathcal{I}} R_{ij} Z_i(t)$$

Clearly, the total network pressure $p(a, Z(t))$ under an allocation a is equal to

$$\sum_{j \in \mathcal{J}} a_j p(j, Z(t)).$$

If j is an idle activity, $p(0, t)$ is set to be 0. The following lemma shows that, in a reversed Leontief network, a maximum pressure policy yields a *separable policy* in the sense that when there are sufficiently many jobs in each constituent buffer, processor k makes decisions based on pressures $p(j, t)$, $j \in \mathcal{J}(k)$, independent of other processors.

Lemma 4.5. *In a reversed Leontief network, for any allocation $a \in \mathcal{E}$, $p(a, Z(t)) = \max_{a' \in \mathcal{E}} p(a', Z(t))$ if and only if*

$$j_k(a) \in \operatorname{argmax}_{j \in \mathcal{J}(k)} p(j, Z(t)) \text{ for all } k \in \mathcal{K}.$$

Proof. Suppose there exist a processor k and an activity $j \in \mathcal{J}(k)$ such that $p(j_k(a), Z(t)) < p(j, Z(t))$. Then we define another allocation $\tilde{a} = a - e_{j_k(a)} + e_j$, where, as before, e_j is the \mathbf{J} -dimensional vector with the j th component equal to 1 and all other components 0. Clearly, \tilde{a} is an integer, feasible allocation, and hence it is extreme. Moreover,

$$p(\tilde{a}, Z(t)) = p(a, Z(t)) - p(j_k(a), Z(t)) + p(j, Z(t)) > p(a, Z(t)).$$

Thus, a cannot be a maximum allocation.

Conversely, suppose that a is an extreme allocation that satisfies $p(j_k(a), Z(t)) \geq p(j, Z(t))$ for all $j \in \mathcal{J}(k)$ and all $k \in \mathcal{K}$. We would like to show that a is a maximum allocation. Let \hat{a} be an extreme maximum allocation, i.e., $\max_{a' \in \mathcal{E}} p(a', Z(t)) = p(\hat{a}, Z(t))$. Then,

$$p(a, Z(t)) = \sum_k p(j_k(a), Z(t)) \geq \sum_k p(j_k(\hat{a}), Z(t)) = p(\hat{a}, Z(t)),$$

which implies that a is a maximum allocation. \square

We end this section by proving Theorem 4.9. We first establish a lemma.

Lemma 4.6. *Define $\hat{\eta}_j(t)$ to be the residual processing time for activity j at time t . Then, for any sample path ω satisfying (4.17) and (4.39),*

$$\lim_{t \rightarrow \infty} \hat{\eta}_j(t)/t = 0.$$

Proof. It is straightforward to show that

$$\hat{\eta}_j(t) \leq \max_{1 \leq \ell \leq S_j(t)} \eta_j(\ell),$$

where $S_j(t)$, as defined in (16), is the number of activity j processing completions in t units of activity j processing time. It follows that

$$\hat{\eta}_j^{1+\epsilon_j}(t) \leq \max_{1 \leq \ell \leq S_j(t)} \eta_j^{1+\epsilon_j}(\ell) \leq \sum_{1 \leq \ell \leq S_j(t)} \eta_j^{1+\epsilon_j}(\ell).$$

Because $S_j(t) \rightarrow \infty$ as $t \rightarrow \infty$, we have

$$\lim_{t \rightarrow \infty} \sum_{1 \leq \ell \leq S_j(t)} \eta_j^{1+\epsilon_j}(\ell)/S_j(t) = m'_j.$$

Therefore,

$$\limsup_{t \rightarrow \infty} \hat{\eta}_j^{1+\epsilon_j}(t)/t \leq \lim_{t \rightarrow \infty} \left(\sum_{1 \leq \ell \leq S_j(t)} \eta_j^{1+\epsilon_j}(\ell)/S_j(t) \right) (S_j(t)/t) = m'_j \mu_j,$$

from which we have

$$\lim_{t \rightarrow \infty} \hat{\eta}_j(t)/t = 0.$$

□

Proof of Theorem 4.9. Let ω be a sample path that satisfies (4.17)–(4.18) and (4.39). Let (\bar{Z}, \bar{T}) be a fluid limit along the sample path. We only need to show that fluid model equation (4.30) is satisfied. The proof is similar to the proof of Lemma 4.1.

First, by the definition of a maximum pressure policy, for any allocation $a \notin \mathcal{E}$, $T^a(t) = 0$ for $t \geq 0$. As a consequence, we have $\sum_{a \in \mathcal{E}} T^a(t) = t$ for $t \geq 0$. It follows that $\sum_{a \in \mathcal{E}} \bar{T}^a(t) = t$ for $t \geq 0$. Recall that, as in (3.2), $p(a, \bar{Z}(t)) = \bar{Z}(t) \cdot Ra$ is the network pressure under allocation a .

Suppose that $a \in \mathcal{E}$ and $p(a, \bar{Z}(t)) < \max_{a' \in \mathcal{E}} p(a', \bar{Z}(t))$. From Assumption 4.1, we can choose an $a^* \in \mathcal{E}$ such that

$$p(a^*, \bar{Z}(t)) = \max_{a' \in \mathcal{E}} p(a', \bar{Z}(t))$$

and the fluid level $\bar{Z}_i(t) > 0$ for all buffers i with $\sum_j a_j^* B_{ji} > 0$. Following the proof of Lemma 4.5, one has that

$$p(j_{\hat{k}}(a), \bar{Z}(t)) < p(j_{\hat{k}}(a^*), \bar{Z}(t))$$

for some processor \hat{k} . Let

$$\hat{j} = j_{\hat{k}}(a), \text{ and } j^* = j_{\hat{k}}(a^*).$$

Now we construct allocation $\hat{a} = a - e_{\hat{j}} + e_{j^*}$. In other words, allocation \hat{a} is exactly the same as allocation a except that processor \hat{k} takes activity j^* instead of \hat{j} . Obviously, $\hat{a} \in \mathcal{E}$ and

$$p(\hat{a}, \bar{Z}(t)) = p(a, \bar{Z}(t)) - p(\hat{j}, \bar{Z}(t)) + p(j^*, \bar{Z}(t)) > p(a, \bar{Z}(t)).$$

We next show that there exists an $\epsilon > 0$ such that for any large number n and each $\tau \in [n(t - \epsilon), n(t + \epsilon)]$,

$$p(a, Z(\tau)) < p(\hat{a}, Z(\tau)), \quad (4.40)$$

$$a \in \mathcal{E}(\tau) \text{ implies that } \hat{a} \in \mathcal{E}(\tau); \text{ namely, } \hat{a} \text{ is a feasible allocation if } a \text{ is.} \quad (4.41)$$

We first assume that $j^* \neq 0$. Denote by $\mathcal{I}(j^*)$ the set of constituency buffers of activity j^* that have potential positive output flows. Namely,

$$\mathcal{I}(j^*) = \{i : B_{j^*i} > 0\}.$$

Then, $\bar{Z}_i(t) > 0$ for all $i \in \mathcal{I}(j^*)$. Since $\bar{p}(\hat{a}, Z(t)) > \bar{p}(a, Z(t))$ and $\min_{i \in \mathcal{I}(j^*)} \bar{Z}_i(t) > 0$, by the continuity of $\bar{\mathbb{X}}(\cdot)$, there exist $\epsilon > 0$ and $\delta > 0$ such that for each $\tau \in [t - \epsilon, t + \epsilon]$ and $i \in \mathcal{I}(j^*)$,

$$p(a, \bar{Z}(\tau)) + \delta \leq p(a^*, \bar{Z}(\tau)) \quad \text{and} \quad \bar{Z}_i(\tau) \geq \delta.$$

Thus, when n is sufficiently large, $p(a, Z(n\tau)) + n\delta/2 \leq p(\hat{a}, Z(n\tau))$ and $Z_i(n\tau) \geq n\delta/2$ for each $i \in \mathcal{I}(j^*)$ and each $\tau \in [t - \epsilon, t + \epsilon]$. Choosing $n > 2\mathbf{J}/\delta$, then for each $\tau \in$

$[n(t - \epsilon), n(t + \epsilon)]$ we have

$$Z_i(\tau) \geq \mathbf{J} \quad \text{for each } i \in \mathcal{I}(j^*). \quad (4.42)$$

Condition (4.42) implies that (4.41) holds, thus proving (4.41) and (4.40). When $j^* = 0$, (4.41) clearly holds for any $\epsilon > 0$ and any n . The proof of (4.40) is identical to the case when $j^* \neq 0$.

Following the definition of the maximum pressure policy and (4.40), if allocation a is not employed at time $n(t - \epsilon)$, it will not be employed during the time interval $[n(t - \epsilon), n(t + \epsilon)]$. If allocation a is employed at time $n(t - \epsilon)$, it will not be deployed during time interval $[n(t - \epsilon) + \hat{\eta}(n(t - \epsilon)), n(t + \epsilon)]$, where $\hat{\eta}(n(t - \epsilon))$ is the longest residual processing time among activities j that are active under allocation a at time $n(t - \epsilon)$. In either case, we have

$$T^a(n(t + \epsilon)) - T^a(n(t - \epsilon)) \leq \max_{j \in \mathcal{J}} \hat{\eta}_j(n(t - \epsilon)).$$

It follows from Lemma 4.6 that

$$\lim_{n \rightarrow \infty} n^{-1}(T^a(n(t + \epsilon)) - T^a(n(t - \epsilon))) = 0.$$

Thus, $\bar{T}^a(t + \epsilon) - \bar{T}^a(t - \epsilon) = 0$, and hence $\dot{\bar{T}}^a(t) = 0$. □

4.7 Applications

In this section, we describe two applications for which the non-processor-splitting, non-preemptive maximum pressure policies are throughput optimal. The first application is to queueing networks with alternative routes. Such a network can be modeled as a stochastic processing network that is *strictly unitary*: each activity is associated with exactly one buffer and it employs exactly one processor. The other application is to networks of data switches. The resulting stochastic processing networks are not unitary. To the best of our knowledge, our maximum pressure policies are the first family of stationary policies, with the buffer level as a state, that are proven to be throughput optimal in these two settings.

Recently, there have been a lot of research activities on parallel server systems or more generally queueing networks with flexible servers [1, 8, 9, 30, 39, 62, 76]. These networks

have been used to model call centers with cross trained operators [4, 5, 31]. We will not get into detailed discussion of these networks except pointing out that maximum pressure policies are also throughput optimal for these systems.

4.7.1 Networks with alternate routing

Consider the queueing network depicted in Figure 4.4. It has 3 servers (represented by circles) processing jobs in 3 buffers (represented by open rectangles). Server i processes jobs exclusively from buffer i , $i = 1, 2, 3$. There are 4 exogenous Poisson arrival processes that are independent. For $i = 1, 2, 3$, jobs from arrival process i goes to buffer i to be processed by server i . Jobs from arrival process 4 can either go to buffer 1 or 2. They are called discretionary jobs. The arrival rate for each process is given in the figure. The processing times for jobs in buffer 2 and 3 are iid having exponential distribution with mean 1. The processing times for jobs in buffer 1 are iid having a general distribution with mean 1. Jobs processed by server 1 go to buffer 3, and jobs processed by Servers 2 and 3 leave the network after processing.

When each server employs a non-idling service policy and processes jobs in an FIFO order, the only decisions that are left for the network are the routing decisions for the discretionary jobs. This network has been studied by Dai and Kim [23] to show that its stability region depends on the processing time distribution for Server 1. Assume that the join-shortest-queue routing policy is employed for the discretionary jobs, and a fair coin is used to break ties. They proved that when the processing time distribution for Server 1

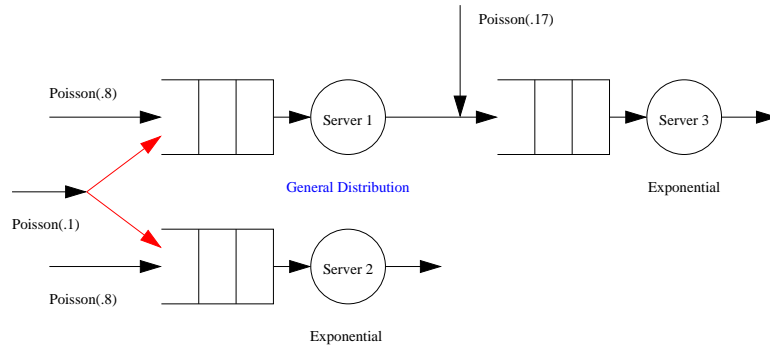


Figure 4.4: A queueing network with alternate routes

is exponential, the network is unstable in the sense that the three dimensional buffer level process is not positive recurrent. They further demonstrated through simulation that the buffer level in buffer 3 grows to infinity linearly as time goes to infinity. When the processing time distribution for Server 1 is hyper-exponential with certain parameters, the network is stable in the sense that a certain Markov chain describing the network is positive recurrent.

Their instability result is not surprising. When processing time distribution for Server 1 is exponential, Servers 1 and 2 are homogeneous. Thus, under the join-shortest-queue routing policy, half of the discretionary jobs go to buffer 1, and these jobs will eventually go to buffer 3. Thus, the traffic intensity of Server 3 is equal to

$$.8 + .17 + .05 = 1.02 > 1,$$

causing the instability of the network. When processing time distribution for Server 1 is hyper-exponential, the processing times have more variability, producing a larger queue in buffer 1 than the one in buffer 2, and hence fewer discretionary jobs joining buffer 1. By choosing a parameter such that the hyper-exponential distribution has high enough variability, the network is actually stable. Having network stability to depend on its processing time distributions is not attractive in practice. For the particular set of network parameters, it can be proven that a round robin routing policy that routes 90% jobs to buffer 2 stabilizes the network. Of course, as the arrival rates change, the percentage in the round robin policy needs to be adjusted as well. Having the policy to depend on the arrival rates is not attractive either when the arrival rates are difficult to be reliably estimated.

We now describe our maximum pressure policies for the network. In turning the queueing network with alternate routes into our stochastic processing network framework, we need to introduce 4 input processors, one for each arrival process. The processing times are equal to interarrival times. Input processor 4 is associated with two input activities, denoted as $(4, 1)$ and $(4, 2)$. Each time the processor completes an activity $(4, i)$ processing, a job goes to buffer i , $i = 1, 2$. The 2 input activities exemplify an extension of the stochastic processing networks in Harrison [36] to allow external inputs with routing capabilities. The resulting stochastic processing network is unitary. The maximum pressure policy in Definition 3.1

amounts to the following: discretionary jobs join the shorter queue among buffers 1 and 2 (with an arbitrary tie-breaking rule); Servers 2 and 3 employ the non-idling service policy; Server 1 stops processing whenever

$$Z_3(t) > Z_1(t). \quad (4.43)$$

By forcing Server 1 to idle when the downstream buffer has more jobs than its buffer, the network is able to propagate its delay to the source nodes, making join-shortest-queue routing policy stable. This example hints the difficulty in finding a pure local policy that is throughput optimal. If, in addition to throughput, other performance measures like average delay is important, one can consider a parameterized family of maximum pressure policies as in Corollary 4.1. In this case, condition (4.43) is replaced by $\gamma_3 Z_3(t) > \gamma_1 Z_1(t) + \theta$ for some positive numbers γ_1, γ_3 and real number θ . For any fixed choice of parameters, the maximum pressure policy is throughput optimal. One can choose parameters to minimize average delay.

Any maximum pressure policy is throughput optimal not only for the network depicted in Figure 4.4 but also for general multiclass queueing networks with alternative routes, a class of networks studied in Laws[46]. Figure 4.5 depicts such a network that was first studied by Laws and Louth [47] and later by Kelly and Laws [42]. There are two types of jobs, horizontal and vertical. For each type of job, there are two arrival activities. Each activity corresponds to a selection of a route that jobs will take. There are 4 service processors represented by circles. Each service processor can process jobs from one of the

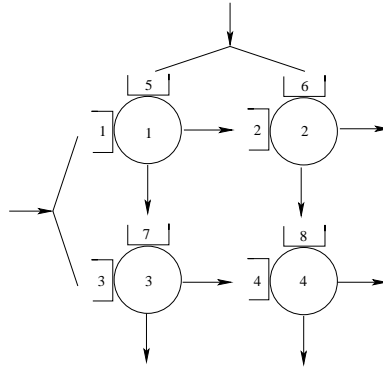


Figure 4.5: A routing network by Laws and Louth

two constituency buffers that are represented by open rectangular boxes. This network can be modeled by a stochastic processing network with a total of 8 service activities and 4 input activities. The example in Figure 4.5 does not have internal routing decisions or probabilistic routing or feedback. A general multiclass queueing network with alternate routes can have all these features. Any non-preemptive, non-processor-splitting maximum pressure policy is throughput optimal in such a network.

Multiclass queueing networks, without routing capabilities, were first introduced by Harrison [33]. They have been used to model re-entrant flows in semiconductor wafer fabrication facilities [17, 20, 48, 72]. Multiclass queueing networks with alternate routes have the potential to model Internet Border Gateway routing [7, 32, 45]. They also have the prospect to serve as road traffic models that support time based or congestion based pricing schemes [29, 70].

4.7.2 Networks of data switches

The Internet is supported by numerous interconnected data switches. A session of data packets from a source to a destination typically traverses a number of switches. The speed at which these switches process the packets influences the propagation delay of the Internet session. Data switches based on an input-queued crossbar architecture are attractive for use in high speed networks [27, 52, 53, 54]. Figure 4.6 depicts a network of 3 input-queued switches. Each switch has 2 input ports and 2 output ports.

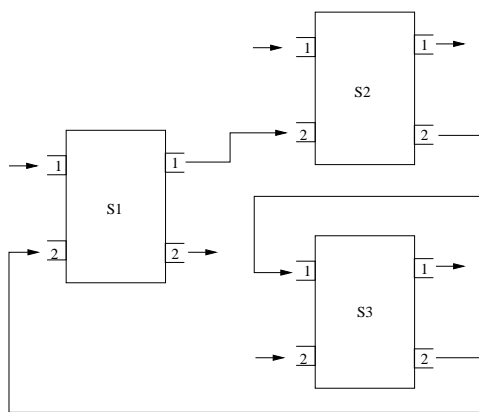


Figure 4.6: A network of input-queued switches

When a session of packets enters the network, it contains information regarding how the packets traverse the network. We assume that these sessions are perpetual (relative to the time window of interest). Such a session is referred to as a *flow* in the rest of this section. Each flow corresponds to a sequence of input and output ports that the packets will traverse. (Each pair of input and output ports is traversed at most once by each flow.) At each input port, we assume there is an infinite capacity buffer that can hold packets to be transmitted. We assume per-flow queueing. That is, each flow maintains a (logical) FIFO queue, called *flow queue*, at each input port. Each flow queue is associated with a pair of input and output ports at a switch. All packets in a flow queue are transmitted to the same output port. On the other hand, a pair of input and output ports may be associated with several flow queues if several flows traverse that input-output pair. A flow queue from flow c at input port ℓ_1 to be transmitted to output port ℓ_2 is denoted by (c, ℓ_1, ℓ_2) . Both the input and output ports should be available to transmit a packet in the flow queue. At a switch, in each time slot, at most one packet is sent from each input port and at most one packet is sent to each output port. Although different switches may employ different time units for a time slot, we assume that each packet takes exactly one unit of time to transmit from a flow queue at the input port to the output port, and then instantaneously to the next flow queue at a downstream input port. The restrictive assumption is mainly for the ease of exposition. The conclusion in this section holds for the general case.

At the beginning of each time slot, each switch needs to find a matching of input and output ports that can simultaneously be active during the slot. It also needs to decide which flow queue to serve at each input port. Define the incidence matrix

$$H_{c,(\ell_1,\ell_2)} = \begin{cases} 1, & \text{if } \ell_1 \text{ and } \ell_2 \text{ are an input-output port pair at} \\ & \text{a switch and flow } c \text{ traverses the pair,} \\ 0, & \text{otherwise.} \end{cases}$$

Assume that α_c is the arrival rate of packets from flow c . It is intuitively clear that for the

network of switches to be stabilizable it is necessary that

$$\sum_{c, \ell_2} \alpha_c H_{c, (\ell_1, \ell_2)} \leq 1 \quad \text{for each input port } \ell_1, \quad (4.44)$$

$$\sum_{c, \ell_1} \alpha_c H_{c, (\ell_1, \ell_2)} \leq 1 \quad \text{for each output port } \ell_2. \quad (4.45)$$

Any scheduling policy that stabilizes the network whenever (4.44) and (4.45) are satisfied is said to be throughput optimal.

For a single switch in isolation, it has been shown that certain policies that are based on maximum weight matchings between input and output ports are throughput optimal, where the weight for an input-output pair is based either on the queue size or the delay [27, 54, 67]. However, Andrews and Zhang [3] provided a network of 8 switches for which the maximum weight policy at each switch is not throughput optimal for the *network*. They further showed that the longest-in-network policy is throughput optimal [3]. A family of Birkhoff-von-Neumann based policies were shown to achieve maximum throughput for networks of switches [51]. These policies use non-local information like arrival rates that may be difficult to estimate in some situations. In the remainder of this section, we are going to show that non-processor-splitting, non-preemptive maximum pressure policies are throughput optimal for networks of switches. The maximum pressure policies are not purely local in that each switch makes scheduling decisions based on the flow queue lengths at the switch and their immediate downstream flow queues.

To state and prove our theorem, we use a stochastic processing network to model the network of input-queued switches. Flow queues serve as buffers in the stochastic processing. There are a total of $\mathbf{L} + 1$ buffers, \mathbf{L} internal buffers for flow queues plus the external buffer 0 modeling the outside world. These internal buffers are indexed by $i = (c, \ell_1, \ell_2)$ representing the flow queue from flow c at input port ℓ_1 destined for output ℓ_2 . Each flow c has an input processor to generate packets from buffer 0 into the network at rate $\alpha_c = 1/m_j$, where j represents the input activity. Each input port is a service processor, and each output is a service processor. Each service activity requires two processors, a pair of input and output ports, to transmit a packet. Service activities are indexed by $j = (c, \ell_1, \ell_2)$. If service activity $j = (c, \ell_1, \ell_2)$ is taken, then a packet of flow c is transmitted from input port ℓ_1 to

output port ℓ_2 at the end of a service completion. When activity $j = (c, \ell_1, \ell_2)$ is active, both input port (processor) ℓ_1 and output port (processor) ℓ_2 are occupied, preventing the transmission of packets from other flow queues through either port ℓ_1 or port ℓ_2 . The total number of service activities is identical to the total number of internal buffers. Clearly, $m_j = 1$ for each service activity j .

Theorem 4.10. *Assume that strong-law-of-large-numbers assumption (4.17) is satisfied for each input activity j . Assume further that traffic conditions (4.44) and (4.45) are satisfied. The network of switches operating under any non-preemptive, non-processor-splitting maximum pressure policy is stable.*

Proof. We first verify that traffic conditions (4.44) and (4.45) are equivalent to the existence of a feasible solution (x, ρ) to the static planning problem (4.3)–(4.6) with $\rho \leq 1$. To see this, one can verify that the input-output matrix R in (3.1) can be written as

$$R_{i,j} = \begin{cases} -\alpha_c, & \text{if input activity } j \text{ generates packets to buffer } i = (c, \ell_1, \ell_2), \\ 1, & \text{if service activity } j \text{ processes packets in buffer } i, \\ -1, & \text{if service activity } j \text{ processes packets that go next to buffer } i. \end{cases}$$

Define an $\mathbf{L} \times \mathbf{L}$ matrix \hat{R} via

$$\hat{R}_{ij} = R_{ij} \quad \text{for each buffer } i \text{ and each service activity } j.$$

Then, we have $\hat{R} = (I - \hat{P})'$, where $\hat{P}_{i,i'} = 1$ if packets in flow queue i go next to flow queue i' , and 0 otherwise.

Let (x, ρ) be a feasible solution to (4.3)–(4.6). Condition (4.5) implies that $x_j = 1$ for each input activity j . Let \hat{x} be the remaining components of x , namely, $\hat{x}_j = x_j$ for each service activity. Condition (4.3) is equivalent to

$$\hat{R}\hat{x} = \lambda,$$

where, for each buffer $i = (c, \ell_1, \ell_2)$, $\lambda_i = \alpha_c$ if flow queue i is the first queue for flow c and $\lambda_i = 0$ otherwise. Since \hat{R} is invertible, we have

$$\hat{x} = \hat{R}^{-1}\lambda.$$

For buffers i and i' , $\hat{R}_{i,i'}^{-1}$ is the number of times that a packet in buffer i' will visit buffer i before it exits the network. Therefore, $\hat{R}_{i,i'}^{-1} = 1$ if $i = i'$ or i is a downstream flow queue of i' , and 0 otherwise. One can verify that $\hat{x}_{(c,\ell_1,\ell_2)} = \alpha_c$ if flow c traverses a pair (ℓ_1, ℓ_2) , and 0 otherwise.

For each input port ℓ_1 , $A_{\ell_1,j} = 1$ if $j = (c, \ell_1, \ell_2)$ is a service activity at the flow queue j , and 0 otherwise. Similarly, for each output port ℓ_2 , $A_{\ell_2,j} = 1$ if $j = (c, \ell_1, \ell_2)$ is a service activity at the flow queue j , and 0 otherwise. Thus, condition (4.4) can be written as

$$\begin{aligned} \sum_{\text{service activities } j} A_{\ell_1,j} \hat{x}_j &= \sum_{c, \ell_2} \hat{x}_{(c,\ell_1,\ell_2)} \leq \rho \quad \text{for each input port } \ell_1, \\ \sum_{\text{service activities } j} A_{\ell_2,j} \hat{x}_j &= \sum_{c, \ell_1} \hat{x}_{(c,\ell_1,\ell_2)} \leq \rho \quad \text{for each output port } \ell_2, \end{aligned}$$

and setting $\rho = 1$ results in the usual traffic intensity conditions (4.44) and (4.45). This proves the equivalence.

For the network of input-queued switches, since each activity is associated with exactly one buffer, it is strict Leontief. Thus, Assumption 4.1 is satisfied. One can further verify that $\mathcal{E} = \mathcal{N}$, and hence condition (4.34)–(4.37) is equivalent to (4.3)–(4.6). Therefore, by Theorem 4.8, any non-processor-splitting maximum pressure policy that allows preemption achieves the optimal throughput. Since the switches operate in time slots, the configurations of the switches can be changed every time slot. As a consequence, the left hand side of equation (4.33) in Section 4.3 is bounded by 1, which implies Lemma 4.1, and hence, Theorems 4.2 and 4.8 still hold. Therefore, the non-preemptive, non-processor-splitting maximum pressure policy is throughput optimal for the network of switches. \square

CHAPTER V

ASYMPTOTIC OPTIMALITY

In this chapter, we investigate the performance of the maximum pressure policies in terms of secondary performance measures. Some secondary performance measures like work-in-process and holding cost can be expressed as functions of the queue length process of a stochastic processing network. In heavy traffic, Reiman’s “snapshot principle” [56, 57, 58] suggests that one can also represent the total delay experienced by an arrival job as a linear combination of the queue lengths seen by that job upon arrival. However, the behavior of the queue length process for a stochastic processing network under any policy is complex. In particular, deriving closed form expressions for performance measures involving these processes is not possible. Therefore, we perform an asymptotic analysis for stochastic processing networks operating under maximum pressure policies. Our asymptotic region is when the network is in heavy traffic; i.e., the offered traffic load is approximately equal to the system capacity.

As a step toward understanding the performance of the maximum pressure policies in terms of general secondary performance measures like holding cost and delay, we establish an asymptotic optimality of the maximum pressure policies for stochastic processing networks with a unique bottleneck in heavy traffic. The optimality is in terms of stochastically minimizing the *workload process* of a stochastic processing network.

The workload process of a stochastic processing network is to be defined in Section 5.1. We will state the main asymptotic optimality result in Section 5.2 and give an outline of the proof of the main theorem in Section 5.3. A key step in the proof is to show that the network process under a maximum pressure policy exhibits a state space collapse, for which we apply Bramson’s framework [12]. In Section 5.4, each fluid model solution under a maximum pressure policy will be shown to exhibit a state space collapse. This will be translated into the state space collapse of the diffusion-scaled network processes in

Section 5.5. At last, in Section 5.6, the state space collapse result will be converted into a heavy traffic limit theorem .

5.1 Workload Process and Complete Resource Pooling

We define the workload process through the following dual LP of the static planning problem (4.2)–(4.6): choose an \mathbf{I} -dimensional vector y and a \mathbf{K} -dimensional vector z so as to

$$\text{maximize} \quad \sum_{k \in \mathcal{K}_I} z_k, \quad (5.1)$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} y_i R_{ij} \leq - \sum_{k \in \mathcal{K}_I} z_k A_{kj}, \text{ for each input activity } j, \quad (5.2)$$

$$\sum_{i \in \mathcal{I}} y_i R_{ij} \leq \sum_{k \in \mathcal{K}_S} z_k A_{kj}, \text{ for each service activity } j, \quad (5.3)$$

$$\sum_{k \in \mathcal{K}_S} z_k = 1, \quad (5.4)$$

$$z_k \geq 0, \text{ for each service processor } k. \quad (5.5)$$

Recall that \mathcal{K}_I is the set of input processors, and \mathcal{K}_S is the set of service processors. Each pair (y, z) that satisfies (5.2)–(5.5) is said to be a *resource pool*. Component y_i is interpreted to be the work dedicated to a unit of buffer i job by the resource pool, and z_k is interpreted to be the relative capacity of processor k , measured in fraction of the service capacity of the resource pool; for each input processor k , the relative capacity z_k is the amount of work generated by input processor k per unit of time. Equality (5.4) ensures that the service capacity of the resource pool equals the sum of service capacities of all service processors. Constraint (5.3) demands that no service activity can accomplish more work than the capacity it consumes. Recall that $-R_{ij}$ is the rate at which input activity j generates buffer i jobs. For each input activity j , constraint (5.2), which can be written as $\sum_{i \in \mathcal{I}} y_i (-R_{ij}) \geq \sum_k z_k A_{kj}$, ensures that the work dedicated to per unit of the activity is no less than that it generates. The objective is to maximize $\sum_{k \in \mathcal{K}_I} z_k$, which is the total amount of work generated from outside by the input processors per unit of time. A service processor k is said to be in the resource pool (y, z) if $z_k > 0$.

A *bottleneck pool* is defined to be an optimal solution (y^*, z^*) to the dual LP (5.1)–(5.5).

Let (ρ^*, x^*) be an optimal solution to the primal LP, the static planning problem (4.2)–(4.6). From the basic duality theory, $\sum_j A_{kj}x_j^* = \rho^*$ for any service processor k with $z_k^* > 0$. It says that all service processors in the bottleneck pool (y^*, z^*) are the busiest servers under any optimal processing plan x^* .

For a bottleneck pool (y^*, z^*) , let $W(t) = y^* \cdot Z(t)$ for $t \geq 0$. Then, $W(t)$ represents the average total work of this bottleneck pool embodied in all jobs that are present at time t in the stochastic processing network. The process $W = \{W(t), t \geq 0\}$ is called the *workload process* of this bottleneck pool. Although the workload process of a non-bottleneck resource pool (y, z) can also be defined by $y \cdot Z(t)$, we will focus on the workload processes of bottleneck pools because bottleneck pools become significantly more important in heavy traffic. In general, the bottleneck pool is not unique. However, we assume all the stochastic processing networks considered in this thesis have a unique bottleneck pool; namely, they satisfy the following *complete resource pooling condition*.

Definition 5.1 (Complete resource pooling condition). A stochastic processing network is said to satisfy the complete resource pooling condition if the corresponding dual static planning problem (5.1)–(5.5) has a nonnegative, unique optimal solution (y^*, z^*) .

For a processing network that satisfies the complete resource pooling condition, we define the *bottleneck workload process*, or simply the workload process, of the stochastic processing network to be the workload process of its unique bottleneck pool.

The (bottleneck) workload process defined here is different from the workload process defined in Harrison and Van Mieghem [34]. Their workload process is multi-dimensional, with some components corresponding to the non-bottleneck pools; it is defined in terms of what they called “reversible displacements”. For the networks where their workload process has dimension one, these two definitions of the workload process are consistent.

Remark. Under certain assumptions including a heavy traffic assumption that requires all servers in the network be critically loaded, Harrison [36] proposed a “canonical” representation of the workload process for stochastic processing networks through a dual LP similar to (5.1)–(5.5). There, basic optimal solutions to the dual LP were chosen as rows of the

workload matrix which was used to define the workload process. Without his heavy traffic assumption, his “canonical” choice of workload matrix would exclude those non-bottleneck servers. In this case, it is not yet clear how to define a “canonical” representation of the workload process to include those nonbottleneck stations. Although, for some network examples like multiclass queueing networks, we can define the workload matrix such that its rows are the basic solutions to the dual LP, more analysis is required for general stochastic processing networks.

5.2 Main Asymptotic Optimality Result

The behavior of the buffer level process and the workload process for a stochastic processing network under any policy is complex. In particular, deriving closed form expressions for performance measures involving these processes is not possible. Therefore, we perform an asymptotic analysis for stochastic processing networks operating under maximum pressure policies. Our asymptotic region is when the network is in heavy traffic; i.e., the offered traffic load is approximately equal to the system capacity. Formally, we consider a sequence of stochastic processing networks indexed by $r = 1, 2, \dots$; as $r \rightarrow \infty$, the traffic intensity ρ^r of the r th network goes to one. We assume that these networks all have the same network topology and primitive increments. In other words, the matrices A and B , and the sequences $(u_j, \phi_i^j : j \in \mathcal{J}, i \in \mathcal{B}_j)$ do not vary with r . However, we allow the processing rates to change with r , and use μ_j^r to denote the processing rate of activity j in the r -th network. Thus, the traffic intensity ρ^r of the r th network is the optimal objective value of the static planning problem (4.2)–(4.6) with the input-output matrix $R^r = (R_{ij}^r)$ given by $R_{ij}^r = \mu_j^r \left(B_{ji} - \sum_{i' \in \mathcal{B}_j} P_{i'i}^j \right)$. We assume the following *heavy traffic assumption* for the rest of this thesis.

Assumption 5.1 (Heavy traffic assumption). There exists a constant $\mu_j > 0$ for each activity $j \in \mathcal{J}$ such that as $r \rightarrow \infty$,

$$\mu_j^r \rightarrow \mu_j, \tag{5.6}$$

and, setting $R = (R_{ij})$ as in (3.1) with μ_j being the limit values in (5.6), the static planning problem (4.2)–(4.6) with parameter (R, A) has a unique optimal solution (ρ^*, x^*) with $\rho^* =$

1. Furthermore, as $r \rightarrow \infty$,

$$r(\rho^r - 1) \rightarrow \theta \quad (5.7)$$

for some constant θ .

We define the *limit network* of the network sequence to be the network that has the same network topology and primitive increments as networks in the sequence, and that has processing rates equal to the limit values μ_j , given in (5.6). Assumption 5.1 basically means that in the limit network there exists only one unique processing plan x^* that can avoid inventory buildups over time, and the busiest service processor is fully utilized under this processing plan. Condition (5.7) requires that the networks' traffic intensities approach to 1 at rate r^{-1} or faster.

The heavy traffic assumption is now quite standard in heavy traffic analysis of queueing networks [13, 19, 24, 73, 75] and stochastic processing networks [6, 36, 39, 34, 76]. However, heavy traffic assumptions in the literature usually assume that, in addition to Assumption 5.1, *all* service processors are fully utilized. The latter assumption, together with the complete resource pooling condition, rules out some common networks such as multiclass queueing networks. In our heavy traffic assumption, only the busiest service processor is required to be critically loaded, and some other service processors are allowed to be under-utilized.

The optimal processing plan x^* given in Assumption 5.1 is referred to as the *nominal processing plan*. Recall that $T_j(t)$ is the cumulative amount of activity j processing time in $[0, t]$ for the limit network, and $T(t)/t$ is the average activity levels over the time span $[0, t]$. To avoid a linear buildup of jobs over time in the limit network, the long-run average rate (or activity level) that activity j is undertaken needs to equal x_j^* , i.e.,

$$\lim_{t \rightarrow \infty} T(t)/t = x^* \text{ almost surely.} \quad (5.8)$$

There should be no linear buildup of jobs under a reasonably “good” policy. A policy is said to be efficient for the limit network if (5.8) holds for the network operating under the policy. Since we consider a sequence of networks, we would like to define an analogous notion of a “good” or efficient policy for the sequence. One can imagine that under a reasonably

“good” policy, when r is large, the average activity levels over long time spans must be very close to the nominal processing plan x^* . To be specific, we define the notion of *asymptotic efficiency* as follows. Let T^r be the cumulative activity level process for the r -th network.

Definition 5.2 (Asymptotic efficiency). Consider a sequence of stochastic processing networks indexed by $r = 1, 2, \dots$, where Assumption 5.1 holds. A policy π is said to be *asymptotically efficient* if and only if under policy π , with probability 1, for each $t \geq 0$,

$$T^r(r^2 t)/r^2 \rightarrow x^* t \quad \text{as } r \rightarrow \infty. \quad (5.9)$$

Equation (5.9) basically says that, under an asymptotically efficient policy, the average activity levels over a time span of order r^2 are very close to the nominal processing plan, so that no linear buildup of jobs will occur over the time span of this order.

Remark. Asymptotic efficiency is closely related to the *throughput optimality* as defined in Chapter 4. Fluid models have been used to prove the throughput optimality of a stochastic processing network operating under a policy. Similarly, the fluid model corresponding to the limit network can be used to prove the asymptotic efficiency of a policy for the sequence of networks that satisfies Assumption 5.1. In particular, one can prove that a policy π is asymptotically efficient if the fluid model of the limit network operating under π is weakly stable.

The following theorem says that a maximum pressure policy is asymptotically efficient for a sequence of networks if the limit network satisfies the EAA assumption, Assumption 4.1.

Theorem 5.1. *Consider a sequence of stochastic processing networks with Assumption 5.1 assumed. If the limit network satisfies the EAA assumption, Assumption 4.1, then a maximum pressure policy is asymptotically efficient.*

The proof of Theorem 5.1 is almost identical to the proof of Theorem 4.2 in Chapter 4, and will be outlined in Appendix B.

Asymptotic efficiency helps to identify reasonably “good” policies, but it is not very discriminating. We would like to demonstrate certain sense of optimality for maximum

pressure policies in terms of secondary performance measures. For this, we will introduce a notion of *asymptotic optimality*. The performance measure for our asymptotic optimality is in terms of the workload process introduced in Section 5.1.

We focus on networks with a single bottleneck pool and assume the following:

Assumption 5.2 (Complete resource pooling). All networks in the sequence and the limit network satisfy the complete resource pooling condition defined in Section 5.1. Namely, the dual static planning problem (5.1)–(5.5) of the r -th network has a nonnegative, unique optimal solution (y^r, z^r) , and the dual static planning problem of the limit network also has a nonnegative, unique optimal solution (y^*, z^*) .

Under Assumption 5.2, we can define the one-dimensional workload process of the r -th network as

$$W^r(t) = y^r \cdot Z^r(t).$$

Remark. In Assumption 5.2, we assume all networks in the sequence satisfy the complete resource pooling condition so that the workload processes W^r can be uniquely defined as in Section 5.1 by the first order network data (R^r, A^r) . This assumption can be removed if one defines the workload process of a network with multiple bottleneck pools to be the workload process of an arbitrarily chosen but prespecified bottleneck pool (with y^r be any given optimal solution to the dual problem). On the other hand, the complete resource pooling condition for the limit network is crucial for our result to hold.

Definition 5.3 (Asymptotic optimality). Consider a sequence of stochastic processing networks indexed by r . An asymptotically efficient policy π^* is said to be *asymptotically optimal* if and only if for any $t > 0, w > 0$, and any asymptotically efficient policy π ,

$$\limsup_{r \rightarrow \infty} \mathbb{P}(W_{\pi^*}^r(r^2 t)/r > w) \leq \liminf_{r \rightarrow \infty} \mathbb{P}(W_{\pi}^r(r^2 t)/r > w), \quad (5.10)$$

where $W_{\pi^*}^r(\cdot)$ and $W_{\pi}^r(\cdot)$ are the workload processes under policies π^* and π , respectively.

Define the diffusion-scaled workload process of the r -th network $\widehat{W}^r = \{\widehat{W}^r(t), t \geq 0\}$ via $\widehat{W}^r(t) = W^r(r^2 t)/r$. Equation (5.10) says that, at every time t , asymptotically, the

diffusion-scaled workload under policy π^* is dominated by that of any other asymptotically efficient policy π in the sense of stochastic ordering.

To state our main theorem, we make a moment assumption on the unitized service times $u_j(\ell)$ and an assumption on the initial queue length processes.

Assumption 5.3. There exists an $\varepsilon_u > 0$ such that, for all j ,

$$\mathbb{E}[(u_j(1))^{2+\varepsilon_u}] < \infty.$$

Assumption 5.3 requires that the unitized service times have finite $2 + \varepsilon_u$ moments. It was introduced by Ata and Kumar [6], and it is stronger than some standard regularity assumptions such as in Bramson [12]. Assumption 5.3 will be used in Section 5.5 to prove a state space collapse result for stochastic processing networks operating under maximum pressure policies.

We also assume that the queue length processes of the stochastic processing networks satisfy the following initial condition. Define the diffusion-scaled queue length process \widehat{Z}^r via $\widehat{Z}^r(t) = Z^r(r^2t)/r$, and let

$$\zeta^r = \frac{y^r}{y^r \cdot y^r}. \quad (5.11)$$

Assumption 5.4 (Initial condition). There exists a random variable w^o such that

$$\widehat{W}^r(0) \rightarrow w^o \text{ in distribution,} \quad (5.12)$$

and

$$|\widehat{Z}^r(0) - \zeta^r \widehat{W}^r(0)| \rightarrow 0 \text{ in probability.} \quad (5.13)$$

Assumption 5.4 holds if the initial queue lengths of the networks are stochastically bounded, namely,

$$\lim_{\tau \rightarrow \infty} \limsup_{r \rightarrow \infty} \mathbb{P}(|Z^r(0)| > \tau) = 0.$$

In this case, $\widehat{Z}^r(0) \rightarrow 0$ in probability.

Theorem 5.2 (Asymptotic Optimality). *Consider a sequence of stochastic processing networks. Assume Assumptions 5.1-5.4 and that the limit network satisfies the EAA assumption. Maximum pressure policies are asymptotically optimal.*

Hereafter, we shall assume Assumptions 5.1–5.4 and that the limit network satisfies the EAA assumption.

Minimizing workload is important even if the ultimate objective is to optimize some other secondary performance measures [6, 8, 36, 64]. For example, in Ata and Kumar [6], the authors demonstrated that their discrete review policies are asymptotically optimal in linear holding costs. They first proved the asymptotic optimality on the workload process. Then the desired asymptotic optimality was achieved because the workload is “appropriately” distributed under the discrete review policies.

Remark. Our asymptotic optimality is defined among the asymptotically efficient policies. One natural question is whether the maximum pressure policies perform better, in terms of equation (5.10), than the policies that are not asymptotically efficient. When the optimal solution (y^*, z^*) to the dual problem (5.1)–(5.5) satisfies $y^* > 0$, it can be proved that the workload process under a maximum pressure policy is stochastically dominated by the workload process under any policy. In fact, one can show that, under a policy that is not asymptotically efficient, at least one component of the diffusion-scaled queue length process $\widehat{Z}^r(t)$ will increase without bound almost surely for any $t > 0$ as $r \rightarrow \infty$. This implies that $\widehat{W}^r(t)$ increases without bound almost surely under the inefficient policy. Although we believe that this is still true when $y_i^* = 0$ for some i , we cannot provide a proof at this point.

5.3 *Outline of the Proof of Asymptotic Optimality*

This section outlines the proof of our main asymptotic optimality theorem, Theorem 5.2. We first derive an asymptotic lower bound on the workload processes under asymptotically efficient policies. Then we state a heavy traffic limit theorem, which implies that this asymptotic lower bound is achieved by the maximum pressure policies. At the end, we outline a proof, for the heavy traffic limit theorem, based on Bramson-Williams’ framework [12, 74, 75].

We first derive an asymptotic lower bound on the workload processes under asymptotically efficient policies. That is, we search for a process W^* such that under any asymptotically efficient policy

$$\liminf_{r \rightarrow \infty} \mathbb{P}(\widehat{W}^r(t) > w) \geq \mathbb{P}(W^*(t) > w), \text{ for all } t \text{ and } w.$$

We begin the analysis by defining a process $Y^r = \{Y^r(t), t \geq 0\}$ for the r -th network via

$$Y^r(t) = (1 - \rho^r)t - y^r \cdot R^r T^r(t). \quad (5.14)$$

Since ρ^r is interpreted as the traffic intensity of the bottleneck pool, for each $t \geq 0$, $\rho^r t$ is interpreted as the average total work contributed to the bottleneck pool from the exogenous arrivals in $[0, t]$, and $(1 - \rho^r)t$ represents the average total work that could have been depleted by time t if the bottleneck pool is never idle. Because of the randomness of the processing times, the idleness of the bottleneck pool will almost surely be incurred over time given the system is not overloaded. Under a service policy and its corresponding activity level process T^r , the average total work that has been depleted by time t is given by

$$y^r \cdot R^r T^r(t) = \sum_{j \in \mathcal{J}_S} T_j^r(t) \sum_{i \in \mathcal{I}} y_i^r R_{ij}^r - \sum_{j \in \mathcal{J}_I} T_j^r(t) \sum_{i \in \mathcal{I}} y_i^r (-R_{ij}^r).$$

Note that, as in Section 5.1, for each service activity $j \in \mathcal{J}_S$, $\sum_{i \in \mathcal{I}} y_i^r R_{ij}^r$ is the average work accomplished by per unit of activity j , and that for each input activity $j \in \mathcal{J}_I$, $\sum_{i \in \mathcal{I}} y_i^r (-R_{ij}^r)$ is the average work generated by per unit of activity j . Therefore, $Y^r(t)$ represents the deviation of the workload depletion in $[0, t]$ from that under the “best” policy. The following lemma says that this deviation does not decrease over time.

Lemma 5.1. *For each r and each sample path, the process Y^r defined in (5.14) is a non-decreasing function with $Y^r(0) = 0$.*

We leave the proof to Appendix B.

Remark. Some special stochastic processing networks, such as multiclass queueing networks and unitary networks, have no control on the input activities. Then, $T_j^r(t)$ is fixed for all $j \in \mathcal{J}_I$ under different policies, and $\sum_{j \in \mathcal{J}_I} T_j^r(t) \sum_{i \in \mathcal{I}} y_i^r (-R_{ij}^r) = \rho^r t$. For these

networks, one gets

$$Y^r(t) = t - \sum_{j \in \mathcal{J}_S} T_j^r(t) \sum_{i \in \mathcal{I}} y_i^r R_{ij}^r,$$

and $Y^r(t)$ is interpreted as the cumulative idle time of the bottleneck pool by time t .

From the flow balance equation (2.5) for the r -th network, we can write the workload process $W^r = y^r \cdot Z^r$ as

$$W^r(t) = W^r(0) + \sum_{i \in \mathcal{I}} y_i^r \sum_{j \in \mathcal{J}} \left(\sum_{i' \in \mathcal{B}_j} \Phi_{i'i}^j(S_j^r(T_j^r(t))) - B_{ji} S_j^r(T_j^r(t)) \right),$$

where $S_j^r(t) = \max\{n : \sum_{\ell=1}^n u_j(\ell) \leq \mu_j^r t\}$.

Let $X^r = W^r - Y^r$. Then one can check that

$$X^r(t) = W^r(0) + \sum_{i \in \mathcal{I}} y_i^r \sum_{j \in \mathcal{J}} \left(\sum_{i' \in \mathcal{B}_j} \Phi_{i'i}^j(S_j^r(T_j^r(t))) - B_{ji} S_j^r(T_j^r(t)) \right) - (1 - \rho^r)t + y^r \cdot R^r T^r(t).$$

We define the following diffusion-scaled processes:

$$\widehat{S}_j^r(t) = r^{-1} [S_j^r(r^2 t) - \mu_j^r r^2 t] \quad \text{for each } j \in \mathcal{J},$$

$$\widehat{\Phi}_{i'i}^j(t) = r^{-1} [\Phi_{i'i}^j(\lfloor r^2 t \rfloor) - P_{i'i}^j r^2 t] \quad \text{for each } j \in \mathcal{J} \text{ and each } i \in \mathcal{B}_j,$$

$$\widehat{X}^r(t) = r^{-1} X^r(r^2 t),$$

$$\widehat{Y}^r(t) = r^{-1} Y^r(r^2 t).$$

Here $\lfloor t \rfloor$ denotes the greatest integer number less than or equal to the real number t .

Then the diffusion-scaled workload process \widehat{W}^r can be written as a sum of two processes

$$\widehat{W}^r(t) = \widehat{X}^r(t) + \widehat{Y}^r(t), \quad (5.15)$$

and

$$\begin{aligned} \widehat{X}^r(t) = \widehat{W}^r(0) + \sum_{i \in \mathcal{I}} y_i^r \sum_{j \in \mathcal{J}} \left(\sum_{i' \in \mathcal{B}_j} \widehat{\Phi}_{i'i}^{j,r}(\bar{\bar{S}}_j^r(\bar{\bar{T}}_j^r(t))) \right. \\ \left. + \left(\sum_{i' \in \mathcal{B}_j} P_{i'i}^j - B_{ji} \right) \widehat{S}_j^r(\bar{\bar{T}}_j^r(t)) \right) - r(1 - \rho^r)t, \end{aligned} \quad (5.16)$$

where

$$\bar{\bar{T}}_j^r(t) = r^{-2} T_j^r(r^2 t) \quad \text{and} \quad \bar{\bar{S}}_j^r(t) = r^{-2} S_j^r(r^2 t).$$

The process \widehat{X}^r depends on the policy only through the fluid-scaled process \bar{T}^r . In fact, from Lemma 4.1 of Dai [21] and (5.9), it follows that, under any asymptotically efficient policy, $\bar{T}^r \Rightarrow x^*(\cdot)$, where $x^*(t) = x^*t$ and x^* is the optimal solution to the static planning problem (4.2)-(4.6) of the limit network. As a consequence, \widehat{X}^r converge to a one-dimensional Brownian motion independent of policies.

Lemma 5.2. *Consider a sequence of stochastic processing networks operating under an asymptotically efficient policy. Then $\widehat{X}^r \Rightarrow X^*$, where X^* is a one-dimensional Brownian motion that starts from w^o given in (5.12), has drift parameter θ given in (5.7), and has variance parameter*

$$\sigma^2 = (y^*)' \left(\sum_{j \in \mathcal{J}} x_j^* \mu_j \sum_{i \in \mathcal{B}_j} \Upsilon^{j,i} \right) y^* + \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} (y_i^*)^2 R_{ij}^2 x_j^* \mu_j \sigma_j^2 \quad (5.17)$$

with $\Upsilon^{j,i}, j \in \mathcal{J}, i \in \mathcal{B}_j$, defined by

$$\Upsilon_{i_1, i_2}^{j,i} = \begin{cases} P_{i, i_1}^j (1 - P_{i, i_2}^j), & \text{if } i_1 = i_2, \\ -P_{i, i_1}^j P_{i, i_2}^j, & \text{if } i_1 \neq i_2. \end{cases}$$

Proof. First, Lemma 4.1 of Dai [21] and (5.9) implies that $\bar{T}^r(\cdot) \Rightarrow x^*(\cdot)$ under any asymptotically efficient policy. Then, the result in the lemma follows from (5.16), the functional central limit theorem for renewal processes (cf. Iglehart and Whitt [41]), the random time change theorem (cf. Billingsley [11] (17.9)), and the continuous mapping theorem (cf. Billingsley [11] Theorem 5.1). Deriving the expression for σ^2 is straightforward but tedious, which is outlined in [33] for multiclass queueing networks, so we will not repeat here. \square

We define the one-dimensional reflection mapping $\psi : \mathbb{D}[0, \infty) \rightarrow \mathbb{D}[0, \infty)$ such that for each $f \in \mathbb{D}[0, \infty)$ with $f(0) \geq 0$,

$$\psi(f)(t) = f(t) - \min(0, \inf_{0 \leq s \leq t} f(s)).$$

Applying diffusion scaling to Lemma 5.1, we know that $\widehat{Y}^r(\cdot)$ is a nonnegative, non-decreasing function, so, from (5.15) and the well-known minimality of the solution of the one-dimensional Skorohod problem (cf. Williams [9] Proposition B.1),

$$\widehat{W}^r(t) \geq \psi(\widehat{X}^r)(t) \quad \text{for every } t \text{ and every sample path;}$$

namely, $\psi(\widehat{X}^r)(t)$ is a pathwise lower bound on \widehat{W}^r . It then follows that

$$\liminf_{r \rightarrow \infty} \mathbb{P}(\widehat{W}^r(t) > w) \geq \limsup_{r \rightarrow \infty} \mathbb{P}(\psi(\widehat{X}^r)(t) > w), \text{ for all } t \text{ and } w.$$

Because $\widehat{X}^r \Rightarrow X^*$, by the continuous mapping theorem, we have

$$\psi(\widehat{X}^r) \Rightarrow W^*,$$

where $W^* = \psi(X^*)$ is a one-dimensional reflecting Brownian motion associated with X^* .

Because $W^*(t)$ has continuous distribution for each t , we have

$$\lim_{r \rightarrow \infty} \mathbb{P}(\psi(\widehat{X}^r)(t) > w) = \mathbb{P}(W^*(t) > w).$$

Therefore,

$$\liminf_{r \rightarrow \infty} \mathbb{P}(\widehat{W}^r(t) > w) \geq \mathbb{P}(W^*(t) > w) \text{ for each } t \text{ and } w.$$

So far, we have shown that W^* is an asymptotic lower bound on the workload processes under asymptotically efficient policies. The following heavy traffic limit theorem ensures that the workload processes under maximum pressure policies converge to W^* . This completes the proof of Theorem 5.2.

Theorem 5.3 (Convergence). *Consider a sequence of stochastic processing networks operating under a maximum pressure policy. Assume Assumptions 5.1-5.4 in Section 5.2 and that the limit network satisfies the EAA assumption. Then*

$$(\widehat{W}^r, \widehat{Z}^r) \Rightarrow (W^*, Z^*) \text{ as } r \rightarrow \infty,$$

where $Z^* = \zeta W^*$ with ζ defined as

$$\zeta = \frac{y^*}{y^* \cdot y^*}. \quad (5.18)$$

Theorem 5.3 also states a form of state space collapse for the network processes in the diffusion limit: The \mathbf{I} -dimensional queue length process is a constant vector multiple of the one-dimensional workload process.

We will apply Bramson-Williams' framework [12, 74, 75] to prove Theorem 5.3. The framework consists of three steps that we will follow in the next three sections: First, in Section 5.4, we will show that any fluid model solution for the stochastic processing networks

under maximum pressure policies exhibit some type of state space collapse, which is stated in Theorem 5.4 in that section. Then, in Section 5.5, we will follow Bramson's approach [12] to translate the state space collapse of the fluid model into a state space collapse result under diffusion scaling which is to be presented in Theorem 5.5. Finally, in Section 5.6, we will convert the state space collapse result to the heavy traffic limit theorem, Theorem 5.3, by applying a perturbed Skorohod mapping theorem from Williams [74].

5.4 State Space Collapse for the Fluid Model

In this section, we show that any fluid model solution under the maximum pressure policy exhibits a form of state space collapse.

Theorem 5.4. *Consider a sequence of stochastic processing networks that satisfy Assumptions 5.1 and 5.2. There exists some finite $\tau_0 > 0$, which depends on just \mathbf{I} , R , and A , such that, for any fluid model solution (\bar{Z}, \bar{T}) under the maximum pressure policy, which satisfies equations (4.7)–(4.11) and (4.13), if $|\bar{Z}(0)| \leq 1$, then*

$$|\bar{Z}(t) - \zeta \bar{W}(t)| = 0, \quad \text{for all } t \geq \tau_0, \quad (5.19)$$

where $\bar{W} = y^* \cdot \bar{Z}$ is the workload process of the fluid model and ζ is given by (5.18). Furthermore, if

$$|\bar{Z}(\tau_1) - \zeta \bar{W}(\tau_1)| = 0, \quad \text{for some } \tau_1 \geq 0,$$

then

$$|\bar{Z}(t) - \zeta \bar{W}(t)| = 0, \quad \text{for all } t \geq \tau_1. \quad (5.20)$$

Theorem 5.4 says that the fluid model under maximum pressure policy exhibits a form of state space collapse: after some finite time τ_0 , the \mathbf{I} -dimensional buffer level process \bar{Z} equals a constant vector multiple of the one-dimensional workload process \bar{W} ; if this happens at time τ_1 , it happens all the time after τ_1 . In particular, if $\bar{Z}(0) = \zeta \bar{W}(0)$, then $\bar{Z}(t) = \zeta \bar{W}(t)$ for all t .

The rest of this section is to prove Theorem 5.4. We first define

$$\bar{Z}^*(t) = \zeta \bar{W}(t),$$

and we shall prove $\bar{Z}(t) - \bar{Z}^*(t) = 0$ for t large enough. Because $\bar{Z}^*(t)$ is the projection of $\bar{Z}(t)$ on y^* , $\bar{Z}(t) - \bar{Z}^*(t)$ is orthogonal to y^* . Therefore, we have the following lemma.

Lemma 5.3. *For any $t \geq 0$,*

$$(\bar{Z}(t) - \bar{Z}^*(t)) \cdot y^* = 0,$$

and for each regular time t ,

$$(\bar{Z}(t) - \bar{Z}^*(t)) \cdot \dot{\bar{Z}}^*(t) = 0.$$

Proof.

$$\bar{Z}^*(t) \cdot y^* = \bar{W}(t)\zeta \cdot y^* = \bar{W}(t) = \bar{Z}(t) \cdot y^*.$$

Because $\dot{\bar{Z}}^*(t) = \zeta \dot{\bar{W}}(t)$,

$$(\bar{Z}(t) - \bar{Z}^*(t)) \cdot \dot{\bar{Z}}^*(t) = \dot{\bar{W}}(t) (\bar{Z}(t) - \bar{Z}^*(t)) \cdot \zeta = 0.$$

□

The following lemma will be used repeatedly in sequel. It follows directly from Lemma A.1 to be presented in Appendix A.

Lemma 5.4. *Suppose (\hat{y}, \hat{z}) is a unique optimal solution to the dual problem (5.1)–(5.5) with objective value ρ . Then \hat{y} is the unique \mathbf{I} -dimensional vector that satisfies*

$$\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} \hat{y}_i R_{ij} a_j = 1, \quad (5.21)$$

and

$$\max_{a \in \mathcal{A}} \hat{y} \cdot Ra = 1 - \rho. \quad (5.22)$$

Define $V = \{Ra : a \in \mathcal{A}\}$. Recall that \mathcal{A} is the set of all possible allocations and the vector Ra is the average rate at which material consumed from all buffers under allocation a , so V is the set of all possible flow rate out of buffers in the limit network. It is obvious that V is a polytope containing the origin because $Rx^* = 0$, where, as before, x^* is the optimal solution to the static planning problem of the limit network. Furthermore, from Lemma 5.4 and the fact that $\rho^* = 1$ for the limit network, we have

$$\max_{v \in V} y^* \cdot v = 0.$$

It says that the outer normal of V at the origin is y^* . Then, V is in a half space separated by the $(\mathbf{I} - 1)$ dimensional hyperplane $V^o = \{v \in \mathbb{R}^I : y^* \cdot v = 0\}$. The hyperplane V^o is orthogonal to y^* and passes the origin, so $V \cap V^o$ is not empty. Furthermore, in the hyperplane V^o , there exists some $(\mathbf{I} - 1)$ dimensional neighborhood of the origin that is a subset of V . This is stated in the following proposition.

Proposition 5.1. *There exists some $\delta > 0$ such that $\{v \in V^o : \|v\| \leq \delta\} \subset V$.*

Proof. First, if the statement is not true, then we can find a $v_0 \in V^o$ such that $\kappa v_0 \notin V$ for all $0 < \kappa \leq 1$ because of the convexity of V . Denote $V_0 = \{\kappa v_0, 0 < \kappa < 1\}$. Because any v in V_0 is not in V , $V_0 \cap V = \emptyset$. It is easy to see that V_0 is open and convex. Therefore there exists a hyperplane separating V and V_0 (cf. Rudin [60] Theorem 3.4). In other words, there exists a vector \hat{y} and a constant b such that $\hat{y} \cdot v \leq b$ for all $v \in V$ and $\hat{y} \cdot v > b$ for all $v \in V_0$. We notice that b must be zero. To see this, first we have $b \geq 0$ because the origin is in V . Moreover, for any $\epsilon > 0$, we can choose κ arbitrarily small such that $\kappa \hat{y} \cdot v_0 < \epsilon$. Because $\kappa v_0 \in V_0$, we have $b < \kappa \hat{y} \cdot v_0 < \epsilon$. This implies $b = 0$, therefore the origin is in the separating hyperplane and $\max_{a \in \mathcal{A}} \hat{y} \cdot Ra = 0$. Obviously, $\hat{y} \neq y^*$ because $y^* \cdot v = 0 > \hat{y} \cdot v$ for $v \in V_0$. We consider two cases:

Case 1: $\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} \hat{y}_i R_{ij} a_j > 0$. For this case, without loss of generality, we select \hat{y} such that $\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} \hat{y}_i R_{ij} a_j = 1$. Then \hat{y} satisfies both (5.21) and (5.22) with $\rho = 1$. On the other hand, from Lemma 5.4, y^* is the unique vector that satisfies both (5.21) and (5.22). This is a contradiction.

Case 2: $\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} \hat{y}_i R_{ij} a_j = 0$. For this case, one can verify that $y^* + \hat{y}$ satisfies both (5.21) and (5.22) with $\rho = 1$. This is also a contradiction.

□

The set $V^* = V \cap V^o$ is the set of all possible flow rates that maximize $y^* \cdot v$; namely, $V^* = \operatorname{argmax}_{v \in V} y^* \cdot v$.

Proof of Theorem 5.4. We consider the Lyapunov function $f(t) = \|\bar{Z}(t) - \bar{Z}^*(t)\|^2$. Let $v(t) = R\dot{\bar{I}}(t)$ denote the net flow rate out of the system at time t (*total departure rate*

minus total arrival rate). Then $\dot{\bar{Z}}(t) = -v(t)$, and we have

$$\dot{f}(t) = 2 (\bar{Z}(t) - \bar{Z}^*(t)) \cdot (\dot{\bar{Z}}(t) - \dot{\bar{Z}}^*(t)) = 2 (\bar{Z}(t) - \bar{Z}^*(t)) \cdot (-v(t)) \quad (5.23)$$

The second equality in (5.23) follows from Lemma 5.3.

Because $\dot{\bar{T}}(t) \in \mathcal{A}$, we have $v(t) \in V$. This implies that $y^* \cdot v(t) \leq 0$ and hence $\bar{Z}^*(t) \cdot v(t) \leq 0$. Furthermore, under maximum pressure policies, it follows from the fluid model equation that $v(t)$ satisfies

$$v(t) \cdot \bar{Z}(t) = \max_{v \in V} \bar{Z}(t) \cdot v.$$

Therefore the last term in (5.23) is bounded from above as follows.

$$2(\bar{Z}(t) - \bar{Z}^*(t)) \cdot (-v(t)) \leq -2\bar{Z}(t) \cdot v(t) = -2 \max_{v \in V} (\bar{Z}(t) \cdot v) \quad (5.24)$$

Since $V^* \subset V$, we have

$$\max_{v \in V} \bar{Z}(t) \cdot v \geq \max_{v \in V^*} (\bar{Z}(t) \cdot v) = \max_{v \in V^*} ((\bar{Z}(t) - \bar{Z}^*(t)) \cdot v). \quad (5.25)$$

The second equality in (5.25) holds because $y^* \cdot v = 0$ for all $v \in V^*$ and $\bar{Z}^*(t) = \zeta \bar{W}(t)$.

If $f(t) > 0$, let

$$v^* = \frac{\delta(\bar{Z}(t) - \bar{Z}^*(t))}{\|\bar{Z}(t) - \bar{Z}^*(t)\|}.$$

Then $\|v^*\| = \delta$ and $y \cdot v^* = 0$. It follows from Proposition 5.1 that $v^* \in V^*$. Therefore

$$\max_{v \in V^*} (\bar{Z}(t) - \bar{Z}^*(t)) \cdot v \geq (\bar{Z}(t) - \bar{Z}^*(t)) \cdot v^* = \delta \|\bar{Z}(t) - \bar{Z}^*(t)\|. \quad (5.26)$$

Combining (5.23)–(5.26), we have

$$\dot{f}(t) \leq -2\delta \|\bar{Z}(t) - \bar{Z}^*(t)\| = -2\delta \sqrt{f(t)}. \quad (5.27)$$

Therefore $f(t) = 0$ for $t \geq \sqrt{f(0)}/\delta$. Set $\tau_0 = \sqrt{\mathbf{I}}/\delta$. Then $f(t) = 0$ for $t \geq \tau_0$, because

$$f(0) = (\bar{Z}(0) - \bar{Z}^*(0)) \cdot (\bar{Z}(0) - \bar{Z}^*(0)) = (\bar{Z}(0) - \bar{Z}^*(0)) \cdot Z(0) \leq \|\bar{Z}(0)\|^2 \leq \mathbf{I}.$$

Here τ_0 depends only on R , A , and \mathbf{I} , because the set V is completely determined by R and A and so is δ .

Equation (5.27) also implies that for any τ_1 , if $f(\tau_1) = 0$ then $f(t) = 0$ for all $t \geq \tau_1$. \square

5.5 State Space Collapse

In this section, we translate the state space collapse result of the fluid model into a state space collapse result under diffusion scaling.

Theorem 5.5 (State Space Collapse). *Consider a sequence of stochastic processing networks operating under a maximum pressure policy. Assume Assumptions 5.1–5.4 and that the limit network satisfies the EAA assumption. Then, for each $T \geq 0$,*

$$\|\widehat{Z}^r(\cdot) - \zeta^r \widehat{W}^r(\cdot)\|_T \rightarrow 0 \text{ in probability.}$$

Recall that ζ^r was given in (5.11) and $\|\cdot\|_T$ is the uniform norm over $[0, T]$. (The readers should not confuse the symbols T and $T(\cdot)$ with one another. We will always include “ (\cdot) ” when dealing with the cumulative activity level process $T(\cdot)$.)

Theorem 5.5 states a form of state space collapse for the diffusion-scaled network process: for large r , the \mathbf{I} -dimensional diffusion-scaled queue length process is essentially a constant vector multiple of the one-dimensional workload process .

The rest of this section applies Bramson’s approach [12] to prove Theorem 5.5. In Bramson’s approach, the following fluid scaling plays an important role in connecting Theorems 5.4 and 5.5: For each $r = 1, 2, \dots$, and $m = 0, 1, \dots$,

$$\begin{aligned} S_j^{r,m}(t) &= \frac{1}{\xi_{r,m}} (S_j^r(rm + \xi_{r,m}t) - S_j^r(rm)), \text{ for each } j \in \mathcal{J}, \\ \Phi_i^{j,r,m}(t) &= \frac{1}{\xi_{r,m}} \left(\Phi_i^j(S_j^r(rm) + \lfloor \xi_{r,m}t \rfloor) - \Phi_i^j(S_j^r(rm)) \right), \text{ for each } j \in \mathcal{J}, i \in \mathcal{B}_j, \\ T_j^{r,m}(t) &= \frac{1}{\xi_{r,m}} (T_j^r(rm + \xi_{r,m}t) - T_j^r(rm)), \text{ for each } j \in \mathcal{J}, \\ Z_i^{r,m}(t) &= \frac{1}{\xi_{r,m}} Z_i^r(rm + \xi_{r,m}t), \text{ for each } i \in \mathcal{I}, \end{aligned}$$

where $\xi_{r,m} = |Z^r(rm)| \vee r$.

Here scaling the processes by $\xi_{r,m}$ ensures $|Z^{r,m}(0)| \leq 1$, which is needed for compactness reasons. And using index (r, m) allows the time scale to expand; we will examine the processes over $[0, L]$ for $m = 0, 1, \dots, \lceil rT \rceil - 1$, where $L > 1$ and $T > 0$ are fixed, so the diffusion-scaled time $[0, T]$ is covered by $\lceil rT \rceil$ fluid scaled time pieces, each with length $L \geq 1$. Here $\lceil t \rceil$ denotes the smallest integer greater than or equal to t .

We outline the proof of Theorem 5.5 as follows. First, in Proposition 5.2, we give a probability estimate on the upper bound of the fluctuation of the stochastic network processes $\mathbb{X}^{r,m}(\cdot) = (T^{r,m}(\cdot), Z^{r,m}(\cdot))$. The estimates on the service processes $S_j^{r,m}(\cdot)$ and the routing processes $\Phi_i^{j,r,m}(\cdot)$ are also given. From Proposition 5.2, a so-called “good” set \mathcal{G}^r of sample paths can be defined, where the processes $\mathbb{X}^{r,m}$ perform nicely for r large enough. On this “good” set, for large enough r , the processes $\mathbb{X}^{r,m}$ can be uniformly approximated by so-called *Lipschitz cluster points*. These cluster points will be shown to be fluid model solutions under the maximum pressure policy. Then because of Theorem 5.4, the state space collapse of the fluid model under the maximum pressure policy, the network processes $\mathbb{X}^{r,m}$ asymptotically have the state space collapse, which then will be translated into the state space collapse for diffusion-scaled processes $\hat{\mathbb{X}}^r$ as r approaches ∞ .

Notice that in Theorem 5.5 the state space collapse of the fluid model does not happen instantaneously after time 0 if the initial state does not exhibit a state space collapse. The fluid-scaled processes $\mathbb{X}^{r,m}$ start from time rm in the original network processes, so, for $m \geq 1$, $\mathbb{X}^{r,m}$ do not automatically have the state space collapse at the initial point, and only the interval $[\tau_0, L]$ can be used for our purpose. However, for $m = 0$, the state space happens at time 0 because of the initial condition (5.13), so the whole interval $[0, L]$ can be used. For this reason, we separate the proof into two parts according to the two intervals in the diffusion-scaled time: $[0, L\xi_{r,0}/r^2]$ and $[\tau_0\xi_{r,0}/r^2, T]$.

Propositions 5.3–5.6 develop a state space collapse on the interval $[\tau_0\xi_{r,0}/r^2, T]$. Proposition 5.3 shows that, on \mathcal{G}^r , the scaled process $\mathbb{X}^{r,m}(\cdot)$ are uniformly close to Lipschitz cluster points for large r . Proposition 5.4 shows that the above cluster points are solutions to the fluid model equations. Propositions 5.3 and 5.4 together with Theorem 5.4, the state space collapse for the fluid model under the maximum pressure policy, imply the state space collapse of the fluid scaled processes $\mathbb{X}^{r,m}(\cdot)$ on \mathcal{G}^r and the interval $[\tau_0, L]$, which is the result of Proposition 5.5. The result is translated into diffusion scaling, and Proposition 5.6 gives a version of state space collapse for the diffusion process $\hat{\mathbb{X}}^r(t)$ on the interval $[\tau_0\xi_{r,0}/r^2, T]$.

The state space collapse on the interval $[0, L\xi_{r,0}/r^2]$ are shown through Propositions 5.7–5.9. The basic idea is the same as described in the preceding paragraph except that now

we only consider the scaled processes with $m = 0$. The corresponding network processes start from time 0, and by assuming the state space collapse happens at time 0, we have stronger result for these type of processes: the state space collapse holds during the whole time interval $[0, L]$ instead of just on $[\tau_0, L]$. In fact, the scaled processes $\mathbb{X}^{r,0}(\cdot)$ are proven to be uniformly close to some cluster points for which the state space collapse starts at time 0. This is stated in Propositions 5.7 and 5.8. In Proposition 5.9, we summarize the state space collapse for fluid scaled process on $[0, L]$ and translate it into the diffusion scale on $[0, L\xi_{r,0}/r^2]$.

The results to be obtained in Propositions 5.6 and 5.9 are actually *multiplicative state space collapse*, as called in Bramson [12]. To obtain the state space collapse result that we stated in Theorem 5.5, we will prove $\xi_{r,m}/r$ are stochastically bounded at the end of this section.

5.5.1 Probability estimates

In this section, we give probability estimates on the service processes $S_j^{r,m}(\cdot)$, the routing processes $\Phi_i^{j,r,m}(\cdot)$, and the upper bound of the fluctuation of the stochastic network processes $\mathbb{X}^{r,m}(\cdot)$.

Proposition 5.2. *Consider a sequence of stochastic processing networks where the moment assumption, Assumption 5.3 is assumed. Fix $\epsilon > 0$, L and T . Then, for large enough r ,*

$$\mathbb{P}(\max_{m < rT} \|S_j^{r,m}(T_j^{r,m}(t)) - \mu_j^r T_j^{r,m}(t)\|_L > \epsilon) \leq \epsilon, \text{ for each } j \in \mathcal{J}, \quad (5.28)$$

$$\mathbb{P}(\max_{m < rT} \|\Phi_i^{j,r,m}(S_j^{r,m}(T_j^{r,m}(t))) - P_i^j \mu_j^r(T_j^{r,m}(t))\|_L > \epsilon) \leq \epsilon, \text{ for each } j \in \mathcal{J} \text{ and } i \in \mathcal{B}_j, \quad (5.29)$$

$$\mathbb{P}(\sup_{0 \leq t_1 \leq t_2 \leq L} |\mathbb{X}^{r,m}(t_2) - \mathbb{X}^{r,m}(t_1)| > N|t_2 - t_1| + \epsilon \text{ for some } m < rT) \leq \epsilon, \quad (5.30)$$

where N is some constant that depends on just the bounds of R^r .

To prove Proposition 5.2, we first need the following lemma.

Lemma 5.5. *Assume that the moment assumption, Assumption 5.3, holds. Then for given T , and each $j \in \mathcal{J}$*

$$u_j^{r,T,max}/r \rightarrow 0 \text{ as } r \rightarrow \infty \text{ with probability 1,} \quad (5.31)$$

where $u_j^{r,T,max} = \max\{u_j(\ell) : 1 \leq \ell \leq S_j^r(r^2T) + 1\}$. Furthermore, for any given $\epsilon > 0$,

$$\mathbb{P}(\|\Phi_i^j(\ell) - P_i^j \ell\|_n \geq \epsilon n) \leq \epsilon/n, \text{ for each } j \in \mathcal{J} \text{ and } i \in \mathcal{B}_j, \text{ and large enough } n, \quad (5.32)$$

and for large enough t ,

$$\mathbb{P}(\|S_j^r(\tau) - \mu_j^r \tau\|_t \geq \epsilon t) \leq \epsilon/t, \text{ for all } j \in \mathcal{J}, \text{ and all } r. \quad (5.33)$$

We delay the proof of Lemma 5.5 to Appendix B, and now we are ready to prove Proposition 5.2.

Proof of Proposition 5.2. The proof here essentially follows the same reasoning as in Propositions 5.1 and 5.2 of Bramson [12]. We first investigate the processes with index m , and then multiply the error bounds by the number of processes in each case, $\lceil rT \rceil$. We first start with (5.28). From (5.31), we have, for large enough r ,

$$\mathbb{P}(u_j^{r,T,max}/r \geq \epsilon) \leq \epsilon/2. \quad (5.34)$$

Denote \mathcal{M}^r to be the complement of the events in (5.34). Then, for large enough r ,

$$\mathbb{P}(\mathcal{M}^r) \geq 1 - \epsilon/2. \quad (5.35)$$

Let τ be the time that the first activity j service completion occurs after time rm . Then from (5.33), for large enough r , we have

$$\mathbb{P}\left(\|S_j^r(rm + \xi_{r,m} T_j^{r,m}(t)) - S_j^r(\tau) - \mu_j^r(rm + \xi_{r,m} T_j^{r,m}(t) - \tau)\|_L \geq \epsilon L \xi_{r,m}\right) \leq \epsilon/Lr;$$

we use the fact that $T_j^{r,m}(t) \leq t$, $\xi_{r,m} \geq r$, and $S_j^r(rm + \xi_{r,m} T_j^{r,m}(t)) - S_j^r(\tau)$ is independent of $\xi_{r,m}$. Because $S_j^r(\tau) = S_j^r(rm) + 1$ and $\tau - rm \leq u_j^{r,T,max}/\mu_j^r$, we have

$$\mathbb{P}\left(\|S_j^r(rm + \xi_{r,m} T_j^{r,m}(t)) - S_j^r(rm) - \mu_j^r \xi_{r,m} T_j^{r,m}(t)\|_L \geq |1 - u_j^{r,T,max}| + \epsilon L \xi_{r,m}\right) \leq \epsilon/Lr.$$

It follows that

$$\mathbb{P}(\|S_j^{r,m}(T_j^{r,m}(t)) - \mu_j^r T_j^{r,m}(t)\|_L \geq 2\epsilon L \mid \mathcal{M}^r) \leq \epsilon/Lr, \text{ for large enough } r. \quad (5.36)$$

Therefore, by multiplying the lower bound by $\lceil rT \rceil$,

$$\mathbb{P}(\max_{m \leq rT} \|S_j^{r,m}(T_j^{r,m}(t)) - \mu_j^r T_j^{r,m}(t)\|_L \geq 2\epsilon L \mid \mathcal{M}^r) \leq \epsilon T.$$

Then enlarging ϵ by a factor of $2(L \vee T)$, we have

$$\mathbb{P}(\max_{m \leq rT} \|S_j^{r,m}(T_j^{r,m}(t)) - \mu_j^r T_j^{r,m}(t)\|_L \geq \epsilon \mid \mathcal{M}^r) \leq \epsilon/2.$$

This, together with (5.35) implies (5.28).

Let μ^{max} be an upper bound on $|\mu^r|$. Then from (5.36), for large enough r ,

$$\mathbb{P}(S_j^{r,m}(T_j^{r,m}(L)) \geq (\mu^{max} + 2)L \mid \mathcal{M}^r) \leq \epsilon/Lr.$$

By (5.32) with $n = (\mu^{max} + 2)L\xi_{r,m}$, we have for each $i \in \mathcal{B}_j$,

$$\begin{aligned} \mathbb{P}\left(\|\Phi_i^{j,r,m}(S_j^{r,m}(T_j^{r,m}(t))) - P_i^j S_j^{r,m}(T_j^{r,m}(t))\|_L > \epsilon(\mu^{max} + 2)L \mid \mathcal{M}^r\right) \\ \leq \epsilon/(\mu^{max} + 2)L\xi_{r,m} + \epsilon/Lr. \end{aligned}$$

Let $P^{max} = \max_{j \in \mathcal{J}, i \in \mathcal{B}_j} |P_i^j|$, then from (5.36), we have

$$\mathbb{P}(\|P_i^j S_j^{r,m}(T_j^{r,m}(t)) - P_i^j \mu_j^r T_j^{r,m}(t)\|_L \geq 2P^{max}\epsilon L \mid \mathcal{M}^r) \leq \epsilon/Lr.$$

It follows that

$$\mathbb{P}\left(\|\Phi_i^{j,r,m}(S_j^{r,m}(T_j^{r,m}(t))) - P_i^j \mu_j^r (T_j^{r,m}(t))\|_L > \epsilon(2P^{max} + \mu^{max} + 2)L \mid \mathcal{M}^r\right) \leq 5\epsilon/2Lr.$$

Enlarging ϵ by a factor of $(2P^{max} + \mu^{max} + 2)L \vee 5T$ and multiplying the error bound by $[rT]$, we have

$$\mathbb{P}\left(\max_{m \leq rT} \|\Phi_i^{j,r,m}(S_j^{r,m}(T_j^{r,m}(t))) - P_i^j \mu_j^r (T_j^{r,m}(t))\|_L > \epsilon \mid \mathcal{M}^r\right) \leq \epsilon/2.$$

Then (5.29) follows.

Now we are going to show (5.30). First, it is easy to see that, for each $j \in \mathcal{J}$ and each r ,

$$T_j^r(t) - T_j^r(s) \leq t - s \text{ for } 0 \leq s \leq t$$

along any sample path. Therefore, the bounds in (5.30) on components T_j is easy to obtain with $N = 1$. For components $Z_i, i \in \mathcal{I}$, scaling (2.5) and applying (5.28) and (5.29) gives

$$\begin{aligned} \mathbb{P}\left(\sup_{0 \leq t_1 \leq t_2 \leq L} |Z_i^{r,m}(t_2) - Z_i^{r,m}(t_1)| \right. \\ \left. > R_{ij}^r |T_j^{r,m}(t_2) - T_j^{r,m}(t_1)| + 2\mathbf{J}(\mathbf{I} + 2)\epsilon \text{ for some } m \leq rT\right) \leq 2\mathbf{J}(\mathbf{I} + 2)\epsilon. \end{aligned}$$

Then (5.30) is obtained by enlarging ϵ in the above inequality by a factor of $2\mathbf{J}(\mathbf{I} + 1)$ and setting $N = 1 \vee \sup_r |R^r|$. \square

Now for each large enough r , we let $\epsilon(r)$ be the smallest ϵ such that (5.28)-(5.30) are satisfied. By Proposition 5.2, it is easy to see that

$$\lim_{r \rightarrow \infty} \epsilon(r) = 0.$$

We consider the complements of the “bad” events given in each of (5.28)-(5.30) with ϵ replaced by $\epsilon(r)$, and denote the intersection of these “good” events by \mathcal{G}^r . Obviously

$$\lim_{r \rightarrow \infty} \mathbb{P}(\mathcal{G}^r) = 1.$$

5.5.2 State space collapse on $[\xi_{r,0}\tau_0/r^2, T]$

We divide the diffusion-scaled time interval $[0, T]$ into two overlapping intervals: $[0, \xi_{r,0}L/r^2]$ and $[\xi_{r,0}\tau_0/r^2, T]$. In this section, we show a state space collapse result on the time interval $[\xi_{r,0}\tau_0/r^2, T]$. The state space collapse on $[0, \xi_{r,0}L/r^2]$ will be presented in the next section.

In order to connect the fluid-scaled processes with the fluid model, we first introduce the notion of *cluster point*. Let $F = \mathbb{D}^d[0, L]$ with $d = \mathbf{I} + \mathbf{J}$, so F is the space of right continuous functions with left limits from $[0, L]$ to $\mathbb{R}^{\mathbf{I}+\mathbf{J}}$. Let $\mathcal{F} = \{F_r\}$ be a sequence of subsets in F . A *cluster point* f of \mathcal{F} is defined to be a point $f \in F$ such that for all $\epsilon > 0$ and $r_0 > 0$, there exists $r \geq r_0$ and $g \in F_r$, with $\|f - g\|_L < \epsilon$. Now, let $F_r, r = 1, 2, \dots$, be certain subsets of F , where all elements f satisfy both

$$|f(0)| \leq 1 \tag{5.37}$$

and

$$|f(t_2) - f(t_1)| \leq N|t_2 - t_1| + \epsilon(r) \text{ for all } t_1, t_2 \in [0, L], \tag{5.38}$$

with constant N chosen as in Proposition 5.2 and $\epsilon(r) \rightarrow 0$ as $r \rightarrow \infty$. The sequence $\{F_r\}$ is said to be *asymptotically Lipschitz*. Let F' denote those $f \in F$ satisfying both (5.37) and

$$|f(t_2) - f(t_1)| \leq N|t_2 - t_1| \text{ for all } t_1, t_2 \in [0, L] \tag{5.39}$$

The following lemma is due to Bramson [12]. We reproduce it here for completeness.

Lemma 5.6 (Bramson [12] Proposition 4.1). *For each $\epsilon > 0$, there exists an r_0 , so that for each $r \geq r_0$ and $g \in F_r$, one has $\|f - g\|_L < \epsilon$ for some cluster point f of \mathcal{F} with $f \in F'$.*

Lemma 5.6 says that, for large r , each element of the function sets in an asymptotically Lipschitz sequence can be uniformly approximated by a cluster point that is Lipschitz continuous.

We set

$$F_g^r = \{\mathbb{X}^{r,m}(\cdot, \omega), \quad m < rT, \omega \in \mathcal{G}^r\} \text{ for each } r,$$

and

$$\mathcal{F}_g = \{F_g^r\}.$$

From the choice of the fluid scale $\xi_{r,m}$ and the definition of $T^{r,m}(\cdot)$, it follows that

$$|\mathbb{X}^{r,m}(0)| \leq 1,$$

so (5.30) in Proposition 5.2 implies that the sequence of sets of scaled stochastic processing network processes $\mathbb{X}^{r,m}(\cdot)$ is asymptotically Lipschitz. Then Lemma 5.6 immediately implies the following Proposition which says that $\mathbb{X}^{r,m}(\cdot)$ are uniformly close to Lipschitz cluster points $\tilde{\mathbb{X}}^{r,m}(\cdot)$ for large r .

Proposition 5.3. *Fix $\epsilon > 0$, L and T , and choose r large enough. Then, for $\omega \in \mathcal{G}^r$ and any $m < rT$,*

$$\|\mathbb{X}^{r,m}(\cdot, \omega) - \tilde{\mathbb{X}}(\cdot)\|_L \leq \epsilon$$

for some cluster point $\tilde{\mathbb{X}}(\cdot)$ of \mathcal{F}_g with $\tilde{\mathbb{X}}(\cdot) \in F'$.

The next proposition says that if the stochastic processing networks operate under a maximum pressure policy, then each cluster point of \mathcal{F}_g is a fluid model solution under the maximum pressure policy and satisfies fluid model equations (4.7)–(4.11) and (4.13).

Proposition 5.4. *Consider a sequence of stochastic processing networks operating under a maximum pressure policy. Assume Assumption 5.1 and that the limit network satisfies the EAA assumption. Fix L and T . Then all cluster points of \mathcal{F}_g are solutions to the fluid model equations (4.7)–(4.11) and (4.13) on $[0, L]$.*

Proof. The idea is to approximate each cluster point of \mathcal{F}_g with some $\mathbb{X}^{r,m}$ on \mathcal{G}^r and show that the equations (4.7)–(4.11) and (4.13) are asymptotically satisfied by $\mathbb{X}^{r,m}$. We will only demonstrate equations (4.7) and (4.13); the others are quite straightforward and can be verified similarly. We first verify equation (4.7). For any $\epsilon > 0$, we can choose large enough r , and appropriate $\omega \in \mathcal{G}^r$ and $m < rT$, such that

$$\|S_j^{r,m}(T_j^{r,m}(t)) - \mu_j^r T_j^{r,m}(t)\|_L \leq \epsilon \text{ for each } j \in \mathcal{J}, \quad (5.40)$$

$$\|\Phi_i^{j,r,m}\left(S_j^{r,m}(T_j^{r,m}(t))\right) - P_i^j \mu_j^r T_j^{r,m}(t)\|_L \leq \epsilon, \text{ for each } j \in \mathcal{J}, i \in \mathcal{B}_j, \quad (5.41)$$

$$\|\tilde{Z}(\cdot) - Z^{r,m}(\cdot)\|_L \leq \epsilon, \quad (5.42)$$

$$\|\tilde{T}(\cdot) - T^{r,m}(\cdot)\|_L \leq \epsilon, \quad (5.43)$$

$$|R^r - R| \leq \epsilon. \quad (5.44)$$

Scaling (2.5) and plugging in the bounds in (5.40) and (5.41), we have

$$|Z^{r,m}(t) - Z^{r,m}(0) - R^r T^{r,m}(t)| \leq 2\epsilon \text{ for each } t \leq L.$$

From (5.43) and (5.44), we have, for each $t \leq L$,

$$|R^r T^{r,m}(t) - R\tilde{T}(t)| \leq |R^r(T^{r,m}(t) - \tilde{T}(t))| + |(R^r - R)\tilde{T}(t)| \leq N\mathbf{J}\epsilon + \epsilon N L \mathbf{J}.$$

Recall that $N \geq \sup_r |R^r|$. It then follows that, for each $t \leq L$,

$$\begin{aligned} |\tilde{Z}(t) - \tilde{Z}(0) - R\tilde{T}(t)| &\leq |\tilde{Z}(t) - Z^{r,m}(t)| + |Z^{r,m}(0) - \tilde{Z}(0)| + |R^r T^{r,m}(t) - R\tilde{T}(t)| \\ &\quad + |Z^{r,m}(t) - Z^{r,m}(0) - R^r T^{r,m}(t)| \\ &\leq (4 + NL\mathbf{J} + N\mathbf{J})\epsilon. \end{aligned}$$

Then equation (4.7) is satisfied by $\tilde{\mathbb{X}}$ because ϵ can be arbitrarily small.

To show equation (4.13), first observe that for any allocation $a \in \mathcal{A}$,

$$\begin{aligned} |p(\tilde{Z}(t), a) - p^r(Z^{r,m}, a)| &= |\tilde{Z}(t) \cdot Ra - Z^{r,m}(t) \cdot R^r a| \\ &\leq |\tilde{Z}(t) \cdot (Ra - R^r a)| + |(\tilde{Z}(t) - Z^{r,m}(t)) \cdot R^r a| \\ &\leq (NL + 1)\mathbf{I}\mathbf{J}\epsilon + \epsilon \mathbf{I}\mathbf{J}N. \end{aligned}$$

Denote \mathcal{E}^* as the set of maximum extreme allocations under buffer size $\tilde{Z}(t)$. Suppose that $a \in \mathcal{E} \setminus \mathcal{E}^*$ and $p(a, \tilde{Z}(t)) < \max_{a' \in \mathcal{E}} p(a', \tilde{Z}(t))$. Because the limit network satisfies the

EAA assumption, we can choose an $a^* \in \mathcal{E}^*$ such that $\tilde{Z}_i(t) > 0$ for each constituent buffer i of a^* . Denote $\mathcal{I}(a^*)$ the set of constituent buffers. Namely,

$$\mathcal{I}(a^*) = \left\{ i : \sum_j a_j^* B_{ji} > 0 \right\}.$$

Then $\tilde{Z}_i(t) > 0$ for all $i \in \mathcal{I}(a^*)$. Since $p(a, \tilde{Z}(t)) < p(a^*, \tilde{Z}(t))$ and $\min_{i \in \mathcal{I}(a^*)} \tilde{Z}_i(t) > 0$, by the continuity of $\tilde{\mathbf{X}}(\cdot)$, there exist $\epsilon_1 > 0$ and $\delta > 0$ such that for each $\tau \in [t - \delta, t + \delta]$ and $i \in \mathcal{I}(a^*)$,

$$p(a, \tilde{Z}(\tau)) + \epsilon_1 \leq p(a^*, \tilde{Z}(\tau)) \quad \text{and} \quad \tilde{Z}_i(\tau) \geq \epsilon_1.$$

For sufficiently large r , we can choose ϵ small enough such that $(NL + N + 1)\mathbf{J}\epsilon \leq \epsilon_1/3$. Then, for all $\tau \in [t - \delta, t + \delta]$,

$$\begin{aligned} p^r(a, Z^{r,m}(\tau)) + \epsilon_1/3 &\leq p^r(a^*, Z^{r,m}(\tau)), \\ Z_i^{r,m}(\tau) &\geq \epsilon_1/2 \quad \text{for each } i \in \mathcal{I}(a^*). \end{aligned}$$

Choosing $r > 2\mathbf{J}/\epsilon_1$, for each $\tau \in [rm + \xi_{r,m}(t - \delta), rm + \xi_{r,m}(t + \delta)]$, we have

$$p^r(a, Z^r(\tau)) < p^r(a^*, Z^r(\tau)), \tag{5.45}$$

$$Z_i^r(\tau) \geq \mathbf{J} \quad \text{for each } i \in \mathcal{I}(a^*). \tag{5.46}$$

Condition (5.46) implies that a^* is a feasible allocation at any time $\tau \in [rm + \xi_{r,m}(t - \delta), rm + \xi_{r,m}(t + \delta)]$, i.e., $a^* \in \mathcal{E}(\tau)$. Following (5.45) and the definition of a (preemptive-resume) maximum pressure policy, the allocation a will not be employed during time interval $[rm + \xi_{r,m}(t - \delta), rm + \xi_{r,m}(t + \delta)]$. Therefore, only the allocations in \mathcal{E}^* will be employed during this interval. For each $a \in \mathcal{E}$, denote $(T^a)^r(t)$ to be the cumulative amount of time allocation a has been employed by time t . It is easy to see that, for each r and all $t \geq 0$, under the maximum pressure policy,

$$\sum_{a \in \mathcal{E}} (T^a)^r(t) = t, \tag{5.47}$$

$$T^r(t) = \sum_{a \in \mathcal{E}} a(T^a)^r(t). \tag{5.48}$$

Then it follows that

$$\begin{aligned}
& \tilde{Z}(t) \cdot \left(R \left[T^r(rm + \xi_{r,m}(t + \delta)) - T^r(rm + \xi_{r,m}(t - \delta)) \right] \right) \\
&= \sum_{a \in \mathcal{E}} (\tilde{Z}(t) \cdot Ra) \left[(T^a)^r(rm + \xi_{r,m}(t + \delta)) - (T^a)^r(rm + \xi_{r,m}(t - \delta)) \right] \\
&= \sum_{a \in \mathcal{E}^*} (\tilde{Z}(t) \cdot Ra) \left[(T^a)^r(rm + \xi_{r,m}(t + \delta)) - (T^a)^r(rm + \xi_{r,m}(t - \delta)) \right] \\
&= \max_{a \in \mathcal{A}} \sum_{a \in \mathcal{E}^*} \left[(T^a)^r(rm + \xi_{r,m}(t + \delta)) - (T^a)^r(rm + \xi_{r,m}(t - \delta)) \right] \\
&= \max_{a \in \mathcal{A}} \sum_{a \in \mathcal{E}} \left[(T^a)^r(rm + \xi_{r,m}(t + \delta)) - (T^a)^r(rm + \xi_{r,m}(t - \delta)) \right] \\
&= 2\xi_{r,m}\delta \max_{a \in \mathcal{A}} \tilde{Z}(t) \cdot Ra.
\end{aligned} \tag{5.49}$$

The second and fourth equalities in (5.49) hold because only allocations in \mathcal{E}^* will be employed during $[rm + \xi_{r,m}(t - \delta), rm + \xi_{r,m}(t + \delta)]$; the third holds because every allocation $a \in \mathcal{E}^*$ has the same network pressure equal to $\max_{a \in \mathcal{A}} \tilde{Z}(t) \cdot Ra$. From (5.49), we have

$$\tilde{Z}(t) \cdot R(T^{r,m}(t + \delta) - T^{r,m}(t - \delta))/2\delta = \max_{a \in \mathcal{A}} \tilde{Z}(t) \cdot Ra.$$

Because ϵ in (5.43) can be arbitrarily small, we have

$$\tilde{Z}(t) \cdot R(\tilde{T}(t + \delta) - \tilde{T}(t - \delta))/2\delta = \max_{a \in \mathcal{A}} \tilde{Z}(t) \cdot Ra,$$

and (4.13) is verified. \square

From Theorem 5.4, every fluid model solution under the maximum pressure policy satisfies (5.19), so Proposition 5.4 implies that any cluster point \tilde{X} of \mathcal{F}_g satisfies (5.19). That is,

$$|\tilde{Z}(t) - \zeta \tilde{W}(t)| = 0 \text{ for } t \geq \tau_0,$$

where $\tilde{W}(\cdot) = y^* \cdot \tilde{Z}(t)$. Because the fluid-scaled stochastic processing network processes can be uniformly approximated by cluster points, it leads to the following proposition.

Proposition 5.5. *Fix L, T and $\epsilon > 0$. For r large enough,*

$$|Z^{r,m}(t) - \zeta^r W^{r,m}(t)| \leq \epsilon \text{ for all } 0 \leq m \leq rT, \tau_0 \leq t \leq L, \omega \in \mathcal{G}^r.$$

To prove Proposition 5.5, we first need $y^r \rightarrow y^*$ as $r \rightarrow \infty$.

Lemma 5.7. *Assume Assumptions 5.1 and 5.2. Then, $(y^r, z^r) \rightarrow (y^*, z^*)$ as $r \rightarrow \infty$.*

The proof of Lemma 5.7 will be provided in Appendix B.

Proof of Proposition 5.5. From Proposition 5.3, for large enough r and each $0 \leq m \leq rT, \omega \in \mathcal{G}^r$, we can find a cluster point \tilde{X} such that

$$\|\tilde{Z}(\cdot) - Z^{r,m}(\cdot)\|_L \leq \epsilon.$$

From Proposition 5.4 and Theorem 5.4, we have

$$|\tilde{Z}(t) - \zeta \tilde{W}(t)| = 0 \text{ for } t \geq \tau_0.$$

Then, for each $t \geq \tau_0$,

$$|Z^{r,m}(t) - \zeta^r W^{r,m}(t)| \leq |Z^{r,m}(t) - \tilde{Z}(t)| + |\zeta \tilde{W}(t) - \zeta^r \tilde{W}(t)| + |\zeta^r \tilde{W}(t) - \zeta^r W^{r,m}(t)|. \quad (5.50)$$

Because $y^r \rightarrow y^*$ as $r \rightarrow \infty$, for large enough r ,

$$|y^r - y^*| < \epsilon, \quad \text{and} \quad |\zeta^r - \zeta| < \epsilon.$$

Let $\kappa = (\sup_r |y^r|) \vee (\sup_r |\zeta^r|)$. Then we have

$$\begin{aligned} |\tilde{W}(t) - W^{r,m}(t)| &\leq |y \cdot \tilde{Z}(t) - y^r \cdot \tilde{Z}(t)| + |y^r \cdot \tilde{Z}(t) - y^r \cdot \tilde{Z}(t)| \\ &\leq \mathbf{I}(\epsilon |\tilde{Z}(t)| + \kappa \epsilon) \leq (L + \kappa) \mathbf{I} \epsilon. \end{aligned}$$

Note that $\tilde{Z}(t) \leq NL$ for all $t \leq L$. One also gets $\tilde{W}(t) \leq \kappa NL$ since $|y^*| \leq \kappa$. From (5.50), we have

$$|Z^{r,m}(t) - \zeta^r W^{r,m}(t)| \leq \epsilon + \epsilon \mathbf{I} \kappa NL + \kappa (NL + \kappa) \mathbf{I} \epsilon.$$

Rechoosing ϵ , the result follows. \square

We need to translate the results in Proposition 5.5 into the state space collapse results under diffusion scaling. First we can express $Z^{r,m}(t)$ by \hat{Z}^r via

$$Z^{r,m}(t) = \frac{r}{\xi_{r,m}} \hat{Z}^r\left(\frac{t\xi_{r,m} + rm}{r^2}\right) = \frac{1}{\bar{\xi}_{r,m}} \hat{Z}^r\left(\frac{t\bar{\xi}_{r,m} + m}{r}\right),$$

where $\bar{\xi}_{r,m} = \xi_{r,m}/r$.

The interval $[\tau_0, L]$ in $Z^{r,m}$ corresponds to the diffusion-scaled time interval $[(m + \bar{\xi}_{r,m}\tau_0)/r, (m + \bar{\xi}_{r,m}L)/r]$, and Proposition 5.5 immediately leads to the following.

Proposition 5.6. Fix $L > 0, T > 0$ and $\epsilon > 0$. For r large enough and each $m < rT$,

$$|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)| \leq \bar{\xi}_{r,m} \epsilon \text{ for all } (m + \bar{\xi}_{r,m} \tau_0)/r \leq t \leq (m + \bar{\xi}_{r,m} L)/r, \omega \in \mathcal{G}^r. \quad (5.51)$$

Proposition 5.6 gives estimates on each small interval for $|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)|$. We shall obtain the estimate on the whole time interval $[0, T]$, and then show $\bar{\xi}_{r,m}$ are stochastically bounded. The following lemma ensures that for large enough L , $L > 3NT$, the small intervals in Proposition 5.6 are overlapping, and therefore the estimate on $[\bar{\xi}_{r,0} \tau_0/r, T]$ is obtained.

Lemma 5.8. For fixed T and large enough r ,

$$\bar{\xi}_{r,m+1} \leq 3N\bar{\xi}_{r,m},$$

for $\omega \in \mathcal{G}^r$ and $m < rT$.

Proof. By the definition of \mathcal{G}^r ,

$$|Z^{r,m}(t_2) - Z^{r,m}(t_1)| \leq N|t_2 - t_1| + 1.$$

for $t_1, t_2 \in [0, L]$ and $m < rT$. Set $t_1 = 0$ and $t_2 = 1/\bar{\xi}_{r,m}$, we have

$$|\widehat{Z}^r((m+1)/r)| - |\widehat{Z}^r(m/r)| \leq N + \bar{\xi}_{r,m}.$$

Therefore,

$$\bar{\xi}_{r,m+1} \leq |\widehat{Z}^r((m+1)/r)| \vee 1 \leq \left(|\widehat{Z}^r(m/r)| + N + \bar{\xi}_{r,m} \right) \vee 1 \leq N + 2\bar{\xi}_{r,m} \leq 3N\bar{\xi}_{r,m}.$$

□

5.5.3 State space collapse on $[0, \bar{\xi}_{r,0} L/r^2]$

Now we shall estimate $|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)|$ on the interval $[0, \bar{\xi}_{r,0} L/r]$. This will be given by the initial condition (5.13) and the result in the second part of Theorem 5.4.

Condition (5.13) implies that

$$|Z^{r,0}(0) - \zeta^r W^{r,0}(0)| \rightarrow 0 \quad \text{in probability.}$$

Then, for each $r > 0$, we let

$$\epsilon_1(r) = \min\{\epsilon : \mathbb{P}(|Z^{r,0}(0) - \zeta^r W^{r,0}(0)| > \epsilon) \leq \epsilon\}.$$

It is easy to see that

$$\lim_{r \rightarrow \infty} \epsilon_1(r) \rightarrow 0.$$

Now let \mathcal{L}^r be the subset of \mathcal{G}^r such that for all events in \mathcal{L}^r

$$|Z^{r,0}(0) - \zeta^r W^{r,0}(0)| > \epsilon_1(r).$$

Obviously, $\lim_{r \rightarrow \infty} \mathbb{P}(\mathcal{L}^r) = 1$.

We define

$$\mathcal{F}_o = \{F_o^r\}$$

with

$$F_o^r = \{\mathbb{X}^{r,0}(\cdot, \omega), \omega \in \mathcal{L}^r\}.$$

Parallel to Proposition 5.3, we have the following proposition which states that \mathcal{F}_o can be asymptotically approximated by cluster points of \mathcal{F}_o .

Proposition 5.7. *Fix $\epsilon > 0$, L and T , and choose r large enough. Then, for $\omega \in \mathcal{L}^r$,*

$$\|\mathbb{X}^{r,0}(\cdot, \omega) - \tilde{\mathbb{X}}(\cdot)\|_L \leq \epsilon$$

for some cluster point $\tilde{\mathbb{X}}(\cdot)$ of \mathcal{F}_o with $\tilde{\mathbb{X}}(\cdot) \in F'$.

Proof. Since both (5.37) and (5.39) are satisfied by $\mathbb{X}^{r,m}$, the result follows from Lemma 5.6. □

Proposition 5.8. *Fix L . Then for any cluster point $\tilde{\mathbb{X}}(\cdot)$ of \mathcal{F}_o ,*

$$\tilde{Z}(t) = \zeta \tilde{W}(t) \quad \text{for } t \leq L.$$

Proof. Since any cluster point of \mathcal{F}_o is automatically a cluster point of \mathcal{F}_g , it satisfies all the fluid model equations. Now we show that

$$|\tilde{Z}(0) - \zeta \tilde{W}(0)| = 0, \tag{5.52}$$

which together with Theorem 5.4 implies the result.

For given $\delta > 0$, one can choose r large enough so that

$$\|\mathbb{X}^{r,0}(\cdot) - \tilde{\mathbb{X}}(\cdot)\|_L \leq \delta,$$

$$|Z^{r,0}(0) - \zeta^r W^{r,0}(0)| \leq \delta,$$

and

$$|\zeta^r - \zeta| \leq \delta.$$

Let $\kappa = \sup_r |y^r|$. Then

$$|W^{r,0}(0)| = |y^r \cdot Z^{r,0}(0)| \leq \mathbf{I}\kappa,$$

and

$$\begin{aligned} |\tilde{W}(0) - W^{r,0}(0)| &= |y^* \cdot \tilde{Z}(0) - y^r \cdot Z^{r,0}(0)| \\ &\leq |y^* \cdot (\tilde{Z}(0) - Z^{r,0}(0)) + (y^* - y^r) \cdot Z^{r,0}(0)| \\ &\leq \mathbf{I}(|y^*| + 1)\delta. \end{aligned}$$

It follows that

$$\begin{aligned} |\tilde{Z}(0) - \zeta \tilde{W}(0)| &\leq |\tilde{Z}(0) - Z^{r,0}(0)| + |Z^{r,0}(0) - \zeta^r W^{r,0}(0)| \\ &\quad + |(\zeta^r - \zeta) W^{r,0}(0)| + |\zeta(W^{r,0}(0) - \tilde{W}(0))| \\ &\leq 2\delta + \mathbf{I}(\mathbf{I}\kappa)\delta + |\zeta|\mathbf{I}(|y| + 1)\delta. \end{aligned}$$

Because δ can be arbitrarily small, we have (5.52) and the result follows. \square

Propositions 5.7 and 5.8 immediately lead to the following proposition, which is parallel to Propositions 5.5 and 5.6.

Proposition 5.9. *Fix L and $\epsilon > 0$. For large enough r ,*

$$|Z^{r,0}(t) - \zeta^r W^{r,0}(t)| \leq \epsilon \text{ for all } 0 \leq t \leq L, \omega \in \mathcal{L}^r,$$

and

$$|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)| \leq \bar{\xi}_{r,0} \epsilon \text{ for all } 0 \leq t \leq \bar{\xi}_{r,0} L, \omega \in \mathcal{L}^r. \quad (5.53)$$

5.5.4 Proof of Theorem 5.5

Propositions 5.6 and 5.9 together give the multiplicative state space collapse of the stochastic processing network processes. To prove the state space collapse result stated in Theorem 5.5, it is enough to prove that $\bar{\xi}_{r,m}$ are stochastically bounded. We first give an upper bound on $\bar{\xi}_{r,m}$ in terms of \widehat{W}^r .

Lemma 5.9. *If $|\widehat{Z}^r(m/r) - \zeta^r \widehat{W}^r(m/r)| \leq 1$, there exists some $\kappa \geq 1$ such that*

$$\bar{\xi}_{r,m} \leq \kappa(\widehat{W}^r(m/r) \vee 1).$$

Proof. We have

$$\bar{\xi}_{r,m} = |\widehat{Z}^r(m/r)| \vee 1 \leq |\zeta^r| \widehat{W}^r(m/r) + 1 \leq 2(|\zeta^r| \widehat{W}^r(m/r) \vee 1)$$

The result follows by choosing $\kappa = 2(\sup_r |\zeta^r| \vee 1)$. \square

The following proposition will be used to derive an upper bound on the oscillation of \widehat{W}^r .

Proposition 5.10. *Consider a sequence of stochastic processing networks operating under maximum pressure policies. There exists $\epsilon_0 > 0$ such that for large enough r , and any $0 \leq t_1 < t_2$, if $\widehat{W}^r(t) \geq \mathbf{J}/(r\epsilon_0)$ and $|\widehat{Z}^r(t)/\widehat{W}^r(t) - \zeta^r| \leq \epsilon_0$ for all $t \in [t_1, t_2]$, then*

$$\widehat{Y}^r(t_2) = \widehat{Y}^r(t_1).$$

Proof. First, we let $0 < \epsilon_0 \leq \zeta_i/3$ for all i with $\zeta_i > 0$. Since $\zeta^r \rightarrow \zeta$ as $r \rightarrow \infty$, for r large enough $|\zeta_i^r - \zeta_i| < \epsilon_0$. Then for all i with $\zeta_i > 0$, we have, for all $t \in [t_1, t_2]$,

$$\widehat{Z}_i^r(t) \geq \widehat{W}^r(t)(\zeta_i^r - \epsilon_0) \geq \widehat{W}^r(t)\epsilon_0 \geq \mathbf{J}/r.$$

Namely, $Z_i^r(\tau) \geq \mathbf{J}$ for all i with $\zeta_i > 0$ and all $\tau \in [r^2 t_1, r^2 t_2]$. Define $\mathcal{E}^* = \operatorname{argmax}_{a \in \mathcal{E}} y^* \cdot Ra$. Because the limit network satisfies the EAA assumption, there exists an allocation $a^* \in \mathcal{E}^*$ such that for each $i \in \mathcal{I}$, $\zeta_i > 0$ if $\sum_j B_{ji} a_j^* > 0$. It follows that $Z_i^r(\tau) \geq \mathbf{J}$ for all $\tau \in [r^2 t_1, r^2 t_2]$ if $\sum_j B_{ji} a_j^* > 0$. This implies that a^* is a feasible allocation during $[r^2 t_1, r^2 t_2]$.

Next, we will show that if ϵ_0 is chosen sufficiently small, only allocations in \mathcal{E}^* can be employed under a maximum pressure policy during $[r^2 t_1, r^2 t_2]$. Let $\epsilon_1 = (\zeta R a^* - \max_{a \in \mathcal{E} \setminus \mathcal{E}^*} \zeta R a)$ and $\kappa_0 = \sup_r \max_{a \in \mathcal{E}} |R^r a|$. We set $\epsilon_0 = \epsilon_1 / (5\kappa_0) \wedge \min\{\zeta_i / 3 : \zeta_i > 0\}$. Furthermore, choose r large enough such that $|(\zeta^r)' R^r - \zeta' R| \leq \kappa_0 \epsilon_0 / \mathbf{J}$. Then for any $a \in \mathcal{E} \setminus \mathcal{E}^*$,

$$\begin{aligned}
& \left(Z^r(t) \cdot R^r a^* - Z^r(t) \cdot R^r a \right) / W^r(t) \\
&= \left(\frac{Z^r(t)}{W^r(t)} \cdot R^r a^* - \zeta^r \cdot R^r a^* \right) + (\zeta^r \cdot R^r a^* - \zeta \cdot R a^*) \\
&+ (\zeta \cdot R a^* - \zeta \cdot R a) + (\zeta \cdot R a - \zeta^r \cdot R^r a) + \left(\zeta^r \cdot R^r a - \frac{Z^r(t)}{W^r(t)} \cdot R^r a \right) \\
&\geq -\epsilon_0 \kappa_0 - \epsilon_0 \kappa_0 + \epsilon_1 - \epsilon_0 \kappa_0 - \epsilon_0 \kappa_0 \geq \epsilon_1 - 4\epsilon_0 \kappa_0 > 0
\end{aligned} \tag{5.54}$$

Equation (5.54) implies that the pressure under allocation a^* is strictly larger than that under any allocation $a \in \mathcal{E} \setminus \mathcal{E}^*$. It follows that only the allocations in \mathcal{E}^* can be employed during $[r^2 t_1, r^2 t_2]$. From Lemma 5.10 stated immediately below this proof, we have, for every $a^* \in \mathcal{E}^*$,

$$y^r R^r a^* = \max_{a \in \mathcal{E}} y^r R^r a^* = 1 - \rho^r.$$

Therefore,

$$\begin{aligned}
y^r \cdot R^r (T^r(r^2 t_2) - T^r(r^2 t_1)) &= \sum_{a \in \mathcal{E}^*} y^r \cdot R^r a ((T^a)^r(r^2 t_2) - (T^a)^r(r^2 t_1)) \\
&= \sum_{a \in \mathcal{E}^*} (1 - \rho^r) ((T^a)^r(r^2 t_2) - (T^a)^r(r^2 t_1)) \\
&= (1 - \rho^r)(r^2 t_2 - r^2 t_1).
\end{aligned} \tag{5.55}$$

This implies $\widehat{Y}(t_2) - \widehat{Y}(t_1) = 0$. □

Lemma 5.10. *For all $a^* \in \mathcal{E}^* = \operatorname{argmax}_{a \in \mathcal{E}} y \cdot R a$,*

$$y^r \cdot R^r a^* = \max_{a \in \mathcal{E}} y^r \cdot R^r a = 1 - \rho^r.$$

The proof of Lemma 5.10 will be provided in Appendix B. We now complete the proof of Theorem 5.5.

Proof of Theorem 5.5. We only need to show that $\bar{\xi}_{r,m}$ are stochastically bounded. Because $\widehat{W}^r(0)$ is stochastically bounded and \widehat{X}^r converge to a Brownian motion, for any $\epsilon > 0$,

there exists some κ_1 and κ_2 and r_1 such that for all $r \geq r_1$

$$\mathbb{P}(\widehat{W}^r(0) \leq \kappa_1) \geq 1 - \epsilon,$$

and

$$\mathbb{P}(\text{Osc}(\hat{X}^r, [0, T]) \leq \kappa_2) \geq 1 - \epsilon,$$

where $\text{Osc}(f, [0, T]) = \sup_{0 \leq s < t \leq T} |f(t) - f(s)|$.

Meanwhile, because

$$|\widehat{Z}^r(0) - \zeta^r \widehat{W}^r(0)| \rightarrow 0 \text{ in probability,}$$

we have

$$\mathbb{P}(|\widehat{Z}^r(0) - \zeta^r \widehat{W}^r(0)| \leq \epsilon) \geq 1 - \epsilon,$$

for r large enough.

Define

$$\mathcal{H}^r = \{\omega : \widehat{W}^r(0) \leq \kappa_1, \text{Osc}(\hat{X}^r, [0, T]) \leq \kappa_2, \text{ and } |\widehat{Z}^r(0) - \zeta^r \widehat{W}^r(0)| \leq \epsilon\}.$$

Then for r large enough,

$$\mathbb{P}(\mathcal{H}^r) \geq 1 - 3\epsilon.$$

Furthermore, we can choose r large enough such that Proposition 5.6 and 5.9 hold with ϵ replaced by $\epsilon/\kappa(\kappa_1 + \kappa_2 + 1)$ and $\mathbb{P}(\mathcal{L}^r) \geq 1 - \epsilon$, where κ is given as in Lemma 5.9. Note that $\mathbb{P}(\mathcal{L}^r) \rightarrow 1$ as $r \rightarrow \infty$.

Denote $\mathcal{N}^r = \mathcal{L}^r \cap \mathcal{H}^r$, then for all r large enough,

$$\mathbb{P}(\mathcal{N}^r) \geq 1 - 4\epsilon.$$

Now we are going to show that if $\epsilon \leq \epsilon_0$, $\bar{\xi}_{r,m} \leq \kappa(\kappa_1 + \kappa_2 + 1)$ on \mathcal{N}^r for all r large enough and $m \leq rT$. In fact, we are going to show the following is true on \mathcal{N}^r for all r large enough and $m \leq rT$:

$$|\widehat{Z}^r(m/r) - \zeta^r \widehat{W}^r(m/r)| \leq \epsilon \quad (5.56)$$

$$\bar{\xi}_{r,m} \leq \kappa(\kappa_1 + \kappa_2 + 1) \quad (5.57)$$

$$|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)| \leq \epsilon \text{ for all } 0 \leq t \leq (m + \bar{\xi}_{r,m}L)/r \quad (5.58)$$

$$\int_0^{(m+\bar{\xi}_{r,m}L)/r} 1_{(\widehat{W}^r(s)>1)} d\widehat{Y}^r(s) = 0. \quad (5.59)$$

This will be shown by induction. When $m = 0$, (5.56) obviously holds on \mathcal{N}^r , and

$$\bar{\xi}_{r,0} = \kappa(\widehat{W}^r(0) \vee 1) \leq \kappa(\kappa_1 \vee 1) \leq \kappa(\kappa_1 + \kappa_2 + 1).$$

Meanwhile, from (5.53), replacing ϵ by $\epsilon/\kappa(\kappa_1 + \kappa_2 + 1)$, we have

$$|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)| \leq \epsilon \text{ for all } t \in [0, \bar{\xi}_{r,0}L/r].$$

Then from Proposition 5.10, with $r \geq 2J/\epsilon_0$, we have

$$\int_0^{\bar{\xi}_{r,0}L/r} 1_{(\widehat{W}^r(s)>1)} d\widehat{Y}^r(s) = 0.$$

Now we assume that (5.56) - (5.59) hold up to m , and we shall show they also hold for $m + 1$. First, (5.58) directly implies (5.56) because $\bar{\xi}_{r,m}L > 1$, and because of (5.59), by Theorem 5.1 in [74], we have

$$\text{Osc}(\widehat{W}^r, [0, (m + \bar{\xi}_{r,m}L)/r]) \leq \text{Osc}(\widehat{X}^r, [0, (m + \bar{\xi}_{r,m}L)/r]) + 1 \leq \kappa_2 + 1.$$

It then follows that

$$\begin{aligned} \bar{\xi}_{r,m+1} &= \kappa(\widehat{W}^r((m+1)/r) \vee 1) \\ &\leq \kappa(\widehat{W}^r(0) + \text{Osc}(\widehat{W}^r, [0, (m+1)/r]) \vee 1) \\ &\leq \kappa(\widehat{W}^r(0) + \text{Osc}(\widehat{W}^r, [0, (m + \bar{\xi}_{r,m}L)/r]) \vee 1) \\ &\leq \kappa(\kappa_1 + \kappa_2 + 1). \end{aligned}$$

And by (5.51),

$$|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)| \leq \epsilon \text{ for all } t \in [(m+1 + \bar{\xi}_{r,m+1}\tau_0)/r, (m+1 + \bar{\xi}_{r,m+1}L)/r].$$

Then because $\bar{\xi}_{r,m}L/r \geq (1 + \bar{\xi}_{r,m+1}\tau_0)/r$, it follows that

$$|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)| \leq \epsilon \text{ for all } t \in [0(m+1 + \bar{\xi}_{r,m+1}L)/r].$$

Then again Proposition 5.10 gives

$$\int_0^{(m+1+\bar{\xi}_{r,m+1}L)/r} 1_{(\widehat{W}^r(s) > 1)} d\widehat{Y}^r(s) = 0.$$

Therefore, we can conclude that $\bar{\xi}_{r,m} \leq \kappa(\kappa_1 + \kappa_2 + 1)$ for all $0 \leq m \leq rT$ which implies

$$\|\widehat{Z}^r(t) - \zeta^r \widehat{W}^r(t)\|_T \leq \epsilon \text{ for all large enough } r.$$

Theorem 5.5 follows because ϵ can be chosen arbitrarily small. \square

5.6 Proof of the Heavy Traffic Limit Theorem

In this section, we will convert the state space collapse results proved in Section 5.5 to the heavy traffic limit theorem, Theorem 5.3. The proof follows from a particular version of the invariance principle of Semimartingale reflecting Brownian motions (SRBMs) developed in Williams [74].

As shown in Section 5.5, for any $T > 0$,

$$\|\widehat{Z}^r(\cdot) - \zeta^r \widehat{W}^r(\cdot)\|_T \rightarrow 0 \text{ in probability.}$$

Now let

$$\varepsilon^r = \widehat{Z}^r - \zeta^r \widehat{W}^r, \text{ and } \delta^r(t) = (|\varepsilon^r(t)| \vee 2\mathbf{J}/r)/\epsilon_0.$$

Then define $\gamma^r(t) = \widehat{W}^r(t) \wedge \delta^r(t)$ and $\tilde{W}^r = \widehat{W}^r - \gamma^r$. It is easy to see that $\tilde{W}^r = (\widehat{W}^r - \delta^r) \vee 0$, and we claim

$$\int_0^\infty \tilde{W}^r(s) d\widehat{Y}^r(s) = 0. \quad (5.60)$$

To show this, it is enough to show that for any $0 \leq t_1 < t_2$, if $\tilde{W}^r(t) > 0$ for all $t \in [t_1, t_2]$, $\widehat{Y}^r(t_2) = \widehat{Y}^r(t_1)$. Suppose $\tilde{W}^r(t) > 0$ for all $t \in [t_1, t_2]$, then $\widehat{W}^r(t) > \delta^r(t)$ for all $t \in [t_1, t_2]$, which implies that $\widehat{W}^r(t) \geq 2\mathbf{J}/(r\epsilon_0)$ and $|\widehat{Z}^r(t)/\widehat{W}^r(t) - \zeta^r| \leq \epsilon_0$. Then by Proposition 5.10, $\widehat{Y}^r(t_2) = \widehat{Y}^r(t_1)$.

With (5.60), we can obtain the result by the following particular version of the invariance principle of SRBMs developed in Williams [74].

Theorem 5.6 (Williams [74] Theorem 4.1). *If the processes $(\widehat{W}^r, \widehat{X}^r, \widehat{Y}^r), r = 1, 2, \dots$, satisfy*

$$\widehat{W}^r = \widehat{X}^r + \widehat{Y}^r,$$

$$\widehat{W}^r = \tilde{W}^r + \gamma^r \text{ where } \tilde{W}^r(t) \geq 0 \text{ for all } t \text{ and } \gamma^r \rightarrow 0 \text{ in probability,}$$

$$\widehat{X}^r \text{ converges in distribution to a Brownian motion } X^*,$$

$$\widehat{Y}^r(0) = 0 \text{ and } \widehat{Y}^r(t) \text{ is non-decreasing with probability 1,}$$

$$\int_0^\infty \tilde{W}^r(s) d\widehat{Y}^r(s) = 0 \text{ with probability 1,}$$

then \widehat{W}^r converge in distribution to a reflecting Brownian motion $W^ = \psi(X^*)$.*

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

The main contribution of this work is to propose the maximum pressure policies for a general class of stochastic processing networks and prove that these policies are both throughput optimal and asymptotically optimal in terms of minimizing the workload process.

The class of stochastic processing networks that we have studied is broad enough to cover a wide range of application fields, including manufacturing systems, service systems, computer systems, and computer communication networks. The maximum pressure policies are attractive in that their implementation uses minimal state information of the network. The deployment of a processor is decided based on the queue lengths in its serviceable buffers and the queue lengths at their immediately downstream buffers. In particular, the decision does not use arrival rate information that is often hard or impossible to estimate reliably.

We have shown that the maximum pressure policies are throughput optimal in the sense that they stabilize the network if the network is stabilizable. The fluid model approach has been adapted to our stochastic processing networks to prove the throughput optimality. For the networks that have a unique bottleneck resource pool, we have proved that the workload processes are stochastically minimized in heavy traffic. A key step in the proof of the asymptotic optimality is to extend Bramson's framework to show a state space collapse result for stochastic processing networks under maximum pressure policies.

The maximum pressure policies are not completely local in that they use immediately downstream buffer information of a processor. Using such information is not an issue in many manufacturing systems, but may be a problem for other systems. Searching for a purely local policy that is throughput optimal remains an open problem.

The maximum pressure policies can maximize system throughput and asymptotically minimize system workload. More practical performance measures include holding cost and

delay. Regarding linear holding costs, it will be shown that for any arbitrarily small ε there is a maximum pressure policy with some parameter such that the holding cost rate under this maximum pressure policy is stochastically smaller than $(1 + \varepsilon)$ times the holding cost rate under any other efficient policy for all time. Evaluating the performance in terms of the delay under maximum pressure policies is another interesting extension. Although it is believed that in heavy traffic, there is a relation between the total delay experienced by an arrival job and the queue length process (or even the workload process), it is not clear how they relate to each other under maximum pressure policies. The other performance measure of interest is load balancing. From the definition of the network pressure, we observe that maximum pressure policies try to maximize the total processing speed meanwhile balancing the buffer levels in the network.

In terms of workload minimization, the results in this work may be generalized in the following two directions. First, the current asymptotic optimality is defined among the asymptotically efficient policies, but one may show that the maximum pressure policies asymptotically minimize the workload among *all* policies. When the workload contributor y^* is strictly positive, one can show that, under a policy that is not asymptotically efficient, at least one component of the diffusion-scaled queue length process $\hat{Z}^r(t)$ will increase without bound almost surely for any $t > 0$ as $r \rightarrow \infty$. This implies that $\widehat{W}^r(t)$ increases without bound almost surely under the inefficient policy. More analysis is needed when some components of y^* equal zero. The other research direction is to relax the complete resource pooling condition. This will allow us to cover the networks with multiple bottlenecks. However, we have multidimensional workload process in this case, and it turns out the queue length process in the diffusion limit cannot be lifted from the workload process. To characterize the performance in the diffusion limit, one will probably need to introduce a generalized version of state space collapse.

APPENDIX A

EQUIVALENT DUAL STATIC PLANNING PROBLEM

In this section we describe an equivalent dual formulation for the static planning problem (4.2)-(4.6). Throughout this section, we will consider an arbitrary stochastic processing network, so the results developed here can be applied to each r -th network in the network sequence we discussed before and they can also be applied to the limit network. The main result of this section is the following lemma.

Lemma A.1. *Suppose ρ is the optimal objective value to the static planning problem (4.2)-(4.6). Then (y^*, z^*) is optimal to the dual problem (5.1)-(5.5) if and only if y^* satisfies*

$$\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_I} y_i^* R_{ij} a_j = -\rho \quad (\text{A.1})$$

and

$$\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} y_i^* R_{ij} a_j = 1, \quad (\text{A.2})$$

and $\{z_k^*, k \in \mathcal{K}_I\}$ and $\{z_k^*, k \in \mathcal{K}_S\}$ are, respectively, optimal solutions to

$$\min \quad - \sum_{k \in \mathcal{K}_I} z_k \quad (\text{A.3})$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} y_i^* R_{ij} \leq - \sum_{k \in \mathcal{K}_I} z_k A_{kj}, \text{ for each input activity } j; \quad (\text{A.4})$$

and

$$\min \quad \sum_{k \in \mathcal{K}_S} z_k \quad (\text{A.5})$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} y_i^* R_{ij} \leq \sum_{k \in \mathcal{K}_S} z_k A_{kj}, \text{ for each service activity } j, \quad (\text{A.6})$$

$$z_k \geq 0, \text{ for each service processor.} \quad (\text{A.7})$$

Proof. For the “only if” part, we assume (x^*, ρ) and (y^*, z^*) are an optimal dual pair for the static planning problem and its dual problem. We will show that y^* satisfies (A.1)-(A.2)

and z^* is optimal to (A.3)-(A.4) and (A.5)-(A.7). We first show that

$$\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_I} y_i^* R_{ij} a_j \geq -\rho, \quad (\text{A.8})$$

and

$$\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} y_i^* R_{ij} a_j \geq 1. \quad (\text{A.9})$$

For this, we construct a feasible allocation a^* with $a_j^* = \begin{cases} x_j^*, & j \in \mathcal{J}_I \\ x_j^*/\rho, & j \in \mathcal{J}_S \end{cases}$. By the complementary slackness on the constraints (5.2) and (5.3), we have

$$\sum_{i \in \mathcal{I}, j \in \mathcal{J}_I} y_i^* R_{ij} x_j^* = - \sum_{k \in \mathcal{K}_I, j \in \mathcal{J}_I} z_k^* A_{kj} x_j^*, \quad (\text{A.10})$$

and

$$\sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} y_i^* R_{ij} x_j^* = \sum_{k \in \mathcal{K}_S, j \in \mathcal{J}_S} z_k^* A_{kj} x_j^*. \quad (\text{A.11})$$

By the complementary slackness on the constraints (4.3) and (4.4), we have

$$\sum_{j \in \mathcal{J}_I, k \in \mathcal{K}_I} z_k^* A_{kj} x_j^* = \sum_{k \in \mathcal{K}_I} z_k = \rho, \quad (\text{A.12})$$

and

$$\sum_{j \in \mathcal{J}_S, k \in \mathcal{K}_S} z_k^* A_{kj} x_j^* = \rho \sum_{k \in \mathcal{K}_S} z_k = \rho. \quad (\text{A.13})$$

The last equality in (A.12) is from the strong duality theorem; the optimal objective value of the dual problem equals the optimal objective value of the primal problem. The last equality in (A.13) follows from (5.4). Then from the definition of a^* and (A.10)-(A.13), it immediately follows that

$$\sum_{i \in \mathcal{I}, j \in \mathcal{J}_I} y_i^* R_{ij} a_j^* = \sum_{i \in \mathcal{I}, j \in \mathcal{J}_I} y_i^* R_{ij} x_j^* = -\rho \quad (\text{A.14})$$

and

$$\sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} y_i^* R_{ij} a_j^* = \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} y_i^* R_{ij} x_j^* / \rho = 1. \quad (\text{A.15})$$

Then (A.8) and (A.9) follow because $a^* \in \mathcal{A}$.

Next we shall show that

$$\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_I} y_i^* R_{ij} a_j \leq -\rho, \quad (\text{A.16})$$

and

$$\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} y_i^* R_{ij} a_j \leq 1. \quad (\text{A.17})$$

For any $a \in \mathcal{A}$, we have

$$\sum_{i \in \mathcal{I}, j \in \mathcal{J}_I} y_i^* R_{ij} a_j \leq - \sum_{k \in \mathcal{K}_I, j \in \mathcal{J}_I} z_k^* A_{kj} a_j = - \sum_{k \in \mathcal{K}_I} z_k^* = -\rho. \quad (\text{A.18})$$

The first inequality above follows from (5.2), and the non-negativity of a ; the second inequality holds since $a \in \mathcal{A}$ and therefore $\sum_{j \in \mathcal{J}_I} A_{kj} a_j = 1$ for each $k \in \mathcal{K}_I$; the third is due to the strong duality theorem. Similarly, we have

$$\sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} y_i^* R_{ij} a_j \leq \sum_{k \in \mathcal{K}_S, j \in \mathcal{J}_S} z_k^* A_{kj} a_j \leq \sum_{k \in \mathcal{K}_S} z_k^* = 1. \quad (\text{A.19})$$

The first inequality above is for (5.3); the second is for $\sum_{j \in \mathcal{J}_S} A_{kj} a_j \leq 1$ and $z_k \geq 0$ for each $k \in \mathcal{K}_S$; and the third is due to (5.4).

Hence y^* satisfies (A.1) and (A.2). To see z^* is an optimal solution to (A.3)-(A.4) and (A.5)-(A.7), we consider the following problems:

$$\max \left\{ \sum_{i \in \mathcal{I}, j \in \mathcal{J}_I} y_i^* R_{ij} a_j : \sum_{j \in \mathcal{J}_I} A_{kj} a_j = 1 \text{ for each input processor } k \right\}, \quad (\text{A.20})$$

and

$$\max \left\{ \sum_{i \in \mathcal{I}, j \in \mathcal{J}_S} y_i^* R_{ij} a_j : \sum_{j \in \mathcal{J}_S} A_{kj} a_j \leq 1 \text{ for each service processor } k \right\}. \quad (\text{A.21})$$

It is easy to see that the above two problems are equivalent to (A.1) and (A.2). Furthermore they are the dual problems of (A.3)-(A.4) and (A.5)-(A.7). This implies that the optimal objective values to (A.3)-(A.4) and (A.5)-(A.7) are $-\rho$ and 1 respectively. Because (y^*, z^*) is an optimal solution to (5.1)-(5.5), (A.4) and (A.6) are satisfied by z^* , $\sum_{k \in \mathcal{K}_I} z_k^* = \rho$, and $\sum_{k \in \mathcal{K}_S} z_k^* = 1$. This implies that z^* is feasible to (A.3)-(A.4) and (A.5)-(A.7) with respective objective values $-\rho$ and 1. Therefore, z^* is optimal to (A.3)-(A.4) and (A.5)-(A.7).

For the “if” part, let (y^*, z^*) be such that y^* satisfies (A.1) and (A.2) and z^* is optimal to (A.3)-(A.4) and (A.5)-(A.7). Because (A.3)-(A.4) and (A.5)-(A.7) are dual problems of the equivalent formulation of (A.1) and (A.2),

$$\sum_{k \in \mathcal{K}_I} z_k^* = \rho, \quad (\text{A.22})$$

and

$$\sum_{k \in \mathcal{K}_S} z_k^* = 1. \quad (\text{A.23})$$

The fact that z^* is feasible to (A.3)-(A.4) and (A.5)-(A.7), together with (A.23), implies that (y^*, z^*) is feasible to the dual problem (5.1)-(5.5). Furthermore, the corresponding objective value is ρ because of (A.22). This implies that (y^*, z^*) is optimal to (5.1)-(5.5). \square

Lemma A.1 immediately implies Lemma 5.4 in Section 5.4 because the LP problem $\max_{a \in \mathcal{A}} y^* \cdot Ra$ can be decomposed into problems (A.20) and (A.21).

APPENDIX B

PROOFS

In this section, we provide the proofs for Lemmas 5.1, 5.5, 5.7, and 5.10, and Theorem 5.1.

Proof of Lemma 5.1. From Lemma 5.4 and Assumption 5.2, we have

$$\max_{a \in \mathcal{A}} y^r \cdot Ra = 1 - \rho^r.$$

On the other hand, $T^r(t)/t \in \mathcal{A}$. Hence $(1 - \rho^r)t \geq y^r \cdot RT^r(t)$, which implies Y^r is nonnegative. Similarly, for any $0 \leq t_1 \leq t_2$, $(T^r(t_2) - T^r(t_1))/(t_2 - t_1) \in \mathcal{A}$, and we have

$$y^r \cdot R(T^r(t_2) - T^r(t_1))/(t_2 - t_1) \leq (1 - \rho),$$

It follows that $Y^r(t_2) - Y^r(t_1) \geq 0$. □

Proof of Lemma 5.5. It is natural to work in a more general setting. Consider an i.i.d. sequence of nonnegative random variables $\{v_\ell, \ell = 1, 2, \dots\}$ with mean $1/\mu_v$. Assume v_ℓ have finite $2 + \epsilon_v$ moments for some $\epsilon_v > 0$. That is, there exists some $\hat{\sigma} < \infty$ such that $\mathbb{E}(v_\ell^{2+\epsilon_v}) = \hat{\sigma}$. Let $V(n) = \sum_{\ell=1}^n v_\ell, n \in \mathbb{Z}^+$. Define the renewal process associated with $V(n)$ as $R(t) = \max\{n : V(n) \leq t\}$. Let $v^{r,T,max} = \max\{v_\ell : 1 \leq \ell \leq R(r^2T) + 1\}$. We first show that

$$v^{r,T,max}/r \rightarrow 0 \text{ with probability 1.} \tag{B.1}$$

By strong law of large numbers, with probability 1,

$$R(t)/t \rightarrow \mu_v \text{ as } t \rightarrow \infty, \tag{B.2}$$

and

$$\sum_{\ell=1}^n v_\ell^{2+\epsilon_v}/n \rightarrow \hat{\sigma} \text{ as } n \rightarrow \infty. \tag{B.3}$$

Choose a sample path such that both (B.2) and (B.3) are satisfied. Then

$$(v^{r,T,max})^{2+\epsilon_v} = \max_{1 \leq \ell \leq R(r^2T)+1} v_\ell^{2+\epsilon_v} \leq \sum_{1 \leq \ell \leq R(r^2T)+1} v_\ell^{2+\epsilon_v}.$$

Because $R(r^2T) \rightarrow \infty$ as $r \rightarrow \infty$, we have

$$\lim_{r \rightarrow \infty} \sum_{1 \leq \ell \leq R(r^2T)+1} v_\ell^{2+\epsilon_v} / (R(r^2T) + 1) = \hat{\sigma}.$$

Therefore,

$$\limsup_{r \rightarrow \infty} \frac{(v^{r,T,max})^{2+\epsilon_v}}{r^2} \leq \lim_{r \rightarrow \infty} \left(\sum_{1 \leq \ell \leq R(r^2T)+1} \frac{v_\ell^{2+\epsilon_v}}{R(r^2T) + 1} \right) \frac{R(r^2T) + 1}{r^2} = \hat{\sigma} \mu_v T.$$

This implies that

$$\lim_{r \rightarrow \infty} \frac{(v^{r,T,max})^{2+\epsilon_v}}{r^{2+\epsilon_v}} = 0.$$

The result follows.

We next show that for any fixed $\epsilon > 0$ and large enough n ,

$$\mathbb{P}(\|V(\ell) - \ell/\mu_v\|_\ell \geq \epsilon n) \leq \epsilon/n. \quad (\text{B.4})$$

Because v_ℓ have finite $2 + \epsilon_v$ moments, one gets

$$\mathbb{E}\left(|V(\ell) - \ell/\mu_v|^{2+\epsilon_v}\right) \leq \kappa_v \ell^{1+\epsilon_v/2} \quad \text{for all } \ell \leq n,$$

where κ_v is a some constant that depends just on $\hat{\sigma}$ and ϵ_v (cf. Ata and Kumar [6] Lemma 8). Then from Chebyshev's inequality, we have, for each $\ell \leq n$,

$$\mathbb{P}(|V(\ell) - \ell/\mu_v| \geq \epsilon n) \leq \kappa_v \ell^{1+\epsilon_v/2} / (\epsilon n)^{2+\epsilon_v} \leq \kappa_v / \epsilon^{2+\epsilon_v} n^{1+\epsilon_v/2}.$$

Choosing n large enough,

$$\mathbb{P}(|V(\ell) - \ell/\mu_v| \geq \epsilon n) \leq \epsilon/n.$$

Let

$$\tau = \min\{\ell : |V(\ell) - \ell/\mu_v| \geq n\epsilon\}.$$

Then, restarting the process at time τ ,

$$\mathbb{P}(|V(n) - n/\mu_v| \leq \epsilon n/2 \mid \tau \leq n) \leq \mathbb{P}(|V(n) - V(\tau) - (n - \tau)/\mu_v| \geq \epsilon n/2 \mid \tau \leq n) \leq \epsilon/2n.$$

On the other hand,

$$\mathbb{P}(|V(n) - n/\mu_v| \leq \epsilon n/2) \geq 1 - \epsilon/2n.$$

It then follows that $\mathbb{P}(\tau \leq n) \leq \epsilon/(2n - \epsilon) \leq \epsilon/n$. This implies the result.

Finally, the inversion of (B.4) gives

$$\mathbb{P}(\|R(s) - \mu_v s\|_t \geq \epsilon t) \leq \epsilon/t \quad \text{for large enough } t. \quad (\text{B.5})$$

Applying (B.1) and (B.5) to the utilized service times $u_j(\ell)$, one gets (5.31) and (5.33) in Lemma 5.5. Using (B.4) for each component of the routing vector $\phi_i^j(\ell)$ yields (5.32). \square

Proof of Lemma 5.7. We first show that $\{(y^r, z^r)\}$ is bounded. Since $z^r \geq 0$, $\sum_{k \in \mathcal{K}_S} z_k^r = 1$, and $\sum_{k \in \mathcal{K}_I} z_k^r = \rho^r$ for all r , we have $|z^r| \leq 1$. To show $\{y^r\}$ is bounded, we consider the following primal-dual pair:

$$\text{minimize} \quad \rho \quad (\text{B.6})$$

$$\text{subject to} \quad Rx \geq be, \quad (\text{B.7})$$

$$\sum_{j \in \mathcal{J}} A_{kj} x_j = 1 \text{ for each input processor } k, \quad (\text{B.8})$$

$$\sum_{j \in \mathcal{J}} A_{kj} x_j \leq \rho \text{ for each service processor } k, \quad (\text{B.9})$$

$$x \geq 0, \quad (\text{B.10})$$

and

$$\text{maximize} \quad \sum_{k \in \mathcal{K}_I} z_k + b \sum_{i \in \mathcal{I}} y_i, \quad (\text{B.11})$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} y_i R_{ij} \leq - \sum_{k \in \mathcal{K}_I} z_k A_{kj}, \text{ for each input activity } j, \quad (\text{B.12})$$

$$\sum_{i \in \mathcal{I}} y_i R_{ij} \leq \sum_{k \in \mathcal{K}_S} z_k A_{kj}, \text{ for each service activity } j, \quad (\text{B.13})$$

$$\sum_{k \in \mathcal{K}_S} z_k = 1, \quad (\text{B.14})$$

$$y \geq 0; \text{ and } z_k \geq 0, \text{ for each service processor } k. \quad (\text{B.15})$$

The dual LP (B.11)–(B.15) is obtained by perturbing the objective function coefficients of the dual static planning (5.1)–(5.5). Because the dual static planning problem (5.1)–(5.5) has a unique optimal solution, for sufficiently small $b > 0$, the optimal solution to the

dual LP (B.11)–(B.15) equals to (y^*, z^*) (cf. Mangasarian [50] Theorem 1). Therefore, the primal problem (B.6)–(B.10) has an optimal solution $(\hat{\rho}, \hat{x})$. Now choose r large enough such that $|R^r \hat{x} - R\hat{x}| < be/2$. Then $R^r \hat{x} \geq be/2$. Consider the problem (B.6)–(B.10) with b replaced by $b/2$ and R replaced by R^r . Because $(\hat{\rho}, \hat{x})$ is a feasible solution, the optimal objective value $\tilde{\rho}^r \leq \hat{\rho}$. The corresponding dual problem of this new LP is the dual problem (B.11)–(B.15) with b in the objective function coefficients replaced by $b/2$ and R replaced by R^r . The optimal objective value of this new dual LP equals $\tilde{\rho}^r \leq \hat{\rho}$. Because (y^r, z^r) is a feasible solution to the new dual LP,

$$\sum_{k \in \mathcal{K}_I} z_k^r + b/2 \sum_{i \in \mathcal{I}} y_i^r \leq \tilde{\rho}^r \leq \hat{\rho}.$$

This implies that $\sum_i y_i^r \leq 2\hat{\rho}/b$ for large enough r , so $\{y^r\}$ is bounded.

Then we only need to show that every convergent subsequence of $\{(y^r, z^r)\}$ converges to (y^*, z^*) . Let (\hat{y}, \hat{z}) be a limit point of any subsequence $\{(y^{r_n}, z^{r_n})\}$, we will verify that (\hat{y}, \hat{z}) is an optimal solution to the dual static planning problem (5.1)–(5.5) of the limiting network. First, we show that they are feasible. Since $\{(y^{r_n}, z^{r_n})\} \rightarrow (\hat{y}, \hat{z})$ as $n \rightarrow \infty$, for any $\epsilon > 0$, for large enough n , $|\hat{y} - y^{r_n}| < \epsilon$, $|\hat{z} - z^{r_n}| < \epsilon$ and $|R - R^{r_n}| < \epsilon$. For each input activity $j \in \mathcal{J}_I$,

$$\begin{aligned} \sum_{i \in \mathcal{I}} \hat{y}_i R_{ij} &\leq \sum_{i \in \mathcal{I}} y_i^{r_n} R_{ij}^{r_n} + \mathbf{I}\epsilon(|\hat{y}| + \sup_r |R^r|) \\ &\leq - \sum_{k \in \mathcal{K}_I} A_{kj} z_k^{r_n} + \mathbf{I}\epsilon(|\hat{y}| + \sup_r |R^r|) \\ &\leq - \sum_{k \in \mathcal{K}_I} A_{kj} \hat{z}_k + \mathbf{I}\epsilon(|\hat{y}| + \sup_r |R^r|) + \mathbf{K}\epsilon \end{aligned} \tag{B.16}$$

Since ϵ can be arbitrarily small, we have

$$\sum_{i \in \mathcal{I}} \hat{y}_i R_{ij} \leq - \sum_{k \in \mathcal{K}_I} A_{kj} \hat{z}_k \text{ for each input activity } j \in \mathcal{J}_I.$$

Similarly, one can verify that

$$\sum_{i \in \mathcal{I}} \hat{y}_i R_{ij} \leq \sum_{k \in \mathcal{K}_S} A_{kj} \hat{z}_k \text{ for each service activity } j \in \mathcal{J}_S,$$

and

$$\sum_{k \in \mathcal{K}_S} \hat{z}_k = 1.$$

Furthermore, because (y^{r_n}, z^{r_n}) are optimal solutions, $\sum_{k \in \mathcal{K}_I} z_k^{r_n} = \rho^{r_n}$. Again we can show that

$$\sum_{k \in \mathcal{K}_I} \hat{z}_k = 1.$$

Therefore, (\hat{y}, \hat{z}) is an optimal solution to the dual problem (5.1)-(5.5) of the limit network. Then by the uniqueness of the optimal solution, we conclude that $(\hat{y}, \hat{z}) = (y^*, z^*)$. Since the subsequence is arbitrary, we have $(y^r, z^r) \rightarrow (y^*, z^*)$ as $r \rightarrow \infty$. \square

Proof of Lemma 5.10. Similar to the proof of Lemma 5.7 above, we can prove that $x^r \rightarrow x^*$ as $r \rightarrow \infty$. From the strict complementary theorem (cf. Wright [77]), every pair of primal and dual LPs has a strict complementary optimal solution if they have optimal solutions. Hence we have the following relations: for the limit network,

$$\sum_{j \in \mathcal{J}_S} A_{kj} x_j^* = 1 \iff z_k^* > 0, \quad \text{for all } k \in \mathcal{K}_S; \quad (\text{B.17})$$

$$\sum_{i \in \mathcal{I}} y_i^* R_{ij} = \sum_{k \in \mathcal{K}_S} A_{kj} z_k^* \iff x_j^* > 0, \quad \text{for all } j \in \mathcal{J}_S; \quad (\text{B.18})$$

and for each r ,

$$z_k^r > 0 \implies \sum_{j \in \mathcal{J}_S} A_{kj} x_j^r = 1 - \rho^r \quad \text{for all } k \in \mathcal{K}_S; \quad (\text{B.19})$$

$$x_j^r > 0 \implies \sum_{i \in \mathcal{I}} y_i^r R_{ij} = \sum_{k \in \mathcal{K}_S} A_{kj} z_k^r, \quad \text{for all } j \in \mathcal{J}_S. \quad (\text{B.20})$$

Since $x^r \rightarrow x^*$ as $r \rightarrow \infty$, for large enough r , $x_j^r > 0$ if $x_j^* > 0$. This, together with (B.18) and (B.20), implies that for each $j \in \mathcal{J}$,

$$y_i^* R_{ij} = \sum_{k \in \mathcal{K}_S} A_{kj} z_k^* \implies \sum_{i \in \mathcal{I}} y_i^r R_{ij} = \sum_{k \in \mathcal{K}_S} A_{kj} z_k^r \text{ for large enough } r. \quad (\text{B.21})$$

Suppose $z_k^* = 0$ for some $k \in \mathcal{K}_S$, then $\sum_{j \in \mathcal{J}_S} A_{kj} x_j^* < 1$. There exists an $\epsilon > 0$ such that $\sum_{j \in \mathcal{J}_S} A_{kj} x_j^* = 1 - \epsilon$. For large enough r , we have $\sum_{j \in \mathcal{J}_S} A_{kj} x_j^r \leq \sum_{j \in \mathcal{J}_S} A_{kj} x_j^* + \epsilon/2 \leq 1 - \epsilon/2$ because $x^r \rightarrow x^*$ as $r \rightarrow \infty$. This implies $z_k^r = 0$ for large enough r by (B.19). Therefore,

$$z_k^* = 0 \implies z_k^r = 0 \text{ for large enough } r \quad (\text{B.22})$$

Because (y^*, z^*) is the optimal solution to the dual static planning problem (5.1)–(5.5), we have for each $a \in \mathcal{E}$,

$$\sum_{j \in \mathcal{J}} \left(\sum_{i \in \mathcal{I}} y_i^* R_{ij} \right) a_j \leq \sum_{j \in \mathcal{J}_S} \left(\sum_{k \in \mathcal{K}_S} A_{kj} z_k^* \right) a_j - \sum_{j \in \mathcal{J}_I} \left(\sum_{k \in \mathcal{K}_I} A_{kj} z_k^* \right) a_j \leq 0 \quad (\text{B.23})$$

Since $a^* \in \mathcal{E}^*$, $y^* \cdot Ra^* = \max_{a \in \mathcal{E}} y^* \cdot Ra = 0$. It follows that both inequalities in (B.23) are equalities for each $a^* \in \mathcal{E}^*$, and therefore,

$$\sum_{i \in \mathcal{I}} y_i^* R_{ij} = \sum_{k \in \mathcal{K}_S} A_{kj} z_k^* \text{ for all } j \in \mathcal{J}_S \text{ with } a_j^* > 0, \quad (\text{B.24})$$

$$\sum_{j \in \mathcal{J}_S} A_{kj} a_j^* = 1 \text{ for all } k \in \mathcal{K}_S \text{ with } z_k^* > 0. \quad (\text{B.25})$$

From (B.21) and (B.24), we have for large enough r ,

$$\sum_{i \in \mathcal{I}} y_i^r R_{ij}^r = \sum_{k \in \mathcal{K}_S} A_{kj} z_k^r \text{ for all } j \in \mathcal{J} \text{ with } a_j^* > 0.$$

Therefore,

$$\sum_{j \in \mathcal{J}} a_j^* \sum_{i \in \mathcal{I}} y_i^r R_{ij}^r = \sum_{j \in \mathcal{J}} a_j^* \sum_{k \in \mathcal{K}_S} A_{kj} z_k^r \text{ for large enough } r. \quad (\text{B.26})$$

From (B.22) and (B.25), it follows that for large enough r ,

$$\sum_{j \in \mathcal{J}_S} a_j^* A_{kj} = 1 \text{ for all } k \in \mathcal{K}_S \text{ with } z_k^r > 0.$$

Therefore,

$$\sum_{k \in \mathcal{K}_S} z_k^r \sum_{j \in \mathcal{J}_S} a_j^* A_{kj} = \sum_{k \in \mathcal{K}_S} z_k^r \text{ for large enough } r. \quad (\text{B.27})$$

It follows from $\sum_{j \in \mathcal{J}_I} a_j^* A_{kj} = 1$ for each $k \in \mathcal{K}_I$, $\sum_{j \in \mathcal{J}_I} z_k^r = \rho^r$, (B.26) and (B.27) that

$$\sum_{j \in \mathcal{J}} a_j^* \sum_{i \in \mathcal{I}} y_i^r R_{ij}^r = \sum_{j \in \mathcal{J}} a_j^* \sum_{k \in \mathcal{K}_S} A_{kj} z_k^r = 1 - \rho^r \text{ for large enough } r. \quad (\text{B.28})$$

Then the result follows from $\max_{a \in \mathcal{E}} y^r \cdot R^r a = 1 - \rho^r$.

□

Proof of Theorem 5.1. We define the scaled process $\bar{\bar{\mathbb{X}}}^r$ via

$$\bar{\bar{\mathbb{X}}}^r(t) = r^{-2} \mathbb{X}(r^2 t) \text{ for each } t \geq 0.$$

Fix a sample path that satisfies the strong law of large numbers for u_j and ϕ_j^i . Let $(\bar{\bar{Z}}, \bar{\bar{T}})$ be a fluid limit of $(\bar{\bar{Z}}^r, \bar{\bar{T}}^r)$ along the sample path. Following the arguments in Section 4.3 such a limit exists and satisfies the fluid model equations (4.7)-(4.11). Under a maximum pressure policy, each fluid limit $(\bar{\bar{Z}}, \bar{\bar{T}})$ also satisfies the fluid model equation (4.13). The justification of fluid model equation (4.13) is similar to Lemma 4.1, with the scaling r replaced by r^2 . Therefore, $(\bar{\bar{Z}}^r, \bar{\bar{T}}^r)$ is a fluid model solution under maximum pressure policy. From Theorem 4.4, the fluid model under the maximum pressure policy is weakly stable, so, under the maximum pressure policy, $(\bar{\bar{Z}}, \bar{\bar{T}})$ satisfies $\bar{\bar{Z}}(t) = 0$ for each $t \geq 0$ given $\bar{\bar{Z}}(0) = 0$. As a consequence, we have for any $t > 0$, $\bar{\bar{T}}(t)/t$ satisfies (4.3)–(4.6) with $\rho = 1$. Because x^* is the unique optimal solution to the static planning problem (4.2)–(4.6) with objective value equal to 1, $\bar{\bar{T}}(t)/t = x^*$ and $\bar{\bar{T}}(t) = x^*t$ for each $t \geq 0$. Since this is true for any fluid limit, we have $\bar{\bar{T}}^r(t) \rightarrow x^*t$ for each t with probability 1, which implies the asymptotic efficiency.

□

REFERENCES

- [1] ANDRADÓTTIR, S., AYHAN, H., and DOWN, D. G., “Dynamic server allocation for queueing networks with flexible servers,” *Operations Research*, vol. 51, pp. 952–968, 2003.
- [2] ANDREWS, M., KUMARAN, K., RAMANAN, K., STOLYAR, A., VIJAYAKUMAR, R., and WHITING, P., “Scheduling in a queueing system with asynchronously varying service rates,” *Probability in the Engineering and Information Sciences*, vol. 18, pp. 191–217, 2004.
- [3] ANDREWS, M. and ZHANG, L., “Achieving stability in networks of input-queued switches,” *IEEE INFOCOM*, pp. 1673–1679, 2001.
- [4] ARMONY, M., *Queueing Networks with Interacting Service Resources*. PhD thesis, Engineering-Economic Systems and Operations Research, Stanford University, 1999.
- [5] ARMONY, M. and BAMBOS, N., “Queueing dynamics and maximal throughput scheduling in switched processing systems,” *Queueing Systems: Theory and Applications*, vol. 44, no. 3, pp. 209–252, 2003.
- [6] ATA, B. and KUMAR, S., “Heavy traffic analysis of open processing networks with complete resource pooling: asymptotic optimality of discrete review policies,” *Annals of Applied Probability*, 2004. To Appear.
- [7] BASU, A., ONG, C.-H. L., RASALA, A., SHEPHERD, F. B., and WILFONG, G., “Route oscillations in I-BGP,” tech. rep., Bell Labs, Lucent Technologies, 2001.
- [8] BELL, S. L. and WILLIAMS, R. J., “Dynamic scheduling of a parallel server system in heavy traffic with complete resource pooling: Asymptotic optimality of a threshold policy,” Preprint, 2004.
- [9] BELL, S. L. and WILLIAMS, R. L., “Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: asymptotic optimality of a threshold policy,” *Annals of Applied Probability*, vol. 11, pp. 608–649, 2001.
- [10] BERTSIMAS, D., PASCHALIDIS, I. C., and TSITSIKLIS, “Optimization of multiclass queueing networks: Polyhedral and nonlinear characterizations of achievable performance,” *Annals of Applied Probability*, vol. 4, pp. 43–75, 1994.
- [11] BILLINGSLEY, P., *Convergence of probability measures*. New York: John Wiley & Sons Inc., second ed., 1999.
- [12] BRAMSON, M., “State space collapse with application to heavy traffic limits for multiclass queueing networks,” *Queueing Systems: Theory and Applications*, vol. 30, pp. 89–148, 1998.
- [13] BRAMSON, M. and DAI, J. G., “Heavy traffic limits for some queueing networks,” *Annals of Applied Probability*, vol. 11, pp. 49–90, 2001.

- [14] BRAMSON, M., "Instability of FIFO queueing networks," *Annals of Applied Probability*, vol. 4, pp. 414–431, 1994.
- [15] BRAMSON, M., "Instability of FIFO queueing networks with quick service times," *Annals of Applied Probability*, vol. 4, pp. 693–718, 1994.
- [16] BRAMSON, M. and WILLIAMS, R. J., "Two workload properties for brownian networks," *Queueing Syst.*, vol. 45, no. 3, pp. 191–221, 2003.
- [17] CHEN, H., HARRISON, J. M., MANDELBAUM, A., VAN ACKERE, A., and WEIN, L. M., "Empirical evaluation of a queueing network model for semiconductor wafer fabrication," *Operations Research*, vol. 36, pp. 202–215, 1988.
- [18] CHEN, H. and YAO, D., "Dynamic scheduling of a multiclass fluid network," *Operations Research*, vol. 41, pp. 1104–1115, 1993.
- [19] CHEN, H. and ZHANG, H., "Diffusion approximations for some multiclass queueing networks with FIFO service disciplines," *Mathematics of Operations Research*, vol. 25, pp. 679–707, 2000.
- [20] CONNORS, D., FEIGIN, G., and YAO, D., "Scheduling semiconductor lines using a fluid network model," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 88–98, 1994.
- [21] DAI, J. G., "On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models," *Annals of Applied Probability*, vol. 5, pp. 49–77, 1995.
- [22] DAI, J. G., "Stability of fluid and stochastic processing networks." MaPhySto Miscellaneous Publication, No. 9, 1999.
- [23] DAI, J. G. and KIM, B., "Stability of Jackson type networks with alternate routes." Preprint, 2003.
- [24] DAI, J. G. and KURTZ, T. G., "A multiclass station with Markovian feedback in heavy traffic," *Mathematics of Operations Research*, vol. 20, pp. 721–742, 1995.
- [25] DAI, J. G. and LIN, W., "Maximum pressure policies in stochastic processing networks," *Operations Research*, 2004. to appear.
- [26] DAI, J. G. and MEYN, S. P., "Stability and convergence of moments for multiclass queueing networks via fluid limit models," *IEEE Transactions on Automatic Control*, vol. 40, pp. 1889–1904, 1995.
- [27] DAI, J. G. and PRABHAKAR, B., "The throughput of data switches with and without speedup," *IEEE INFOCOM*, pp. 556–564, 2000.
- [28] DANTZIG, G. B., "The programming of interdependent activities: mathematical models," in *Activity Analysis of Production and Allocation* (Koopmans, T. C., ed.), (New York), pp. 19–32, John Wiley and Sons, 1951.
- [29] FAWCETT, J. and ROBINSON, P., "Adaptive routing for road traffic," *IEEE Computer Graphics and Applications*, vol. 20, pp. 46–53, 2000.

- [30] GANS, N. and VAN RYZIN, G., “Optimal control of a multiclass, flexible queueing system,” *Operations Research*, vol. 45, pp. 677–693, 1997.
- [31] GANS, N., KOOLE, G., and MANDELBAUM, A., “Telephone call centers: a tutorial and literature review,” *Manufacturing and Service Operations Management*, vol. 5, pp. 79–141, 2003.
- [32] GAO, L. and REXFORD, J., “Stable Internet routing without global coordination,” *IEEE/ACM Transactions on Networking*, vol. 9, pp. 681–692, 2001.
- [33] HARRISON, J. M., “Brownian models of queueing networks with heterogeneous customer populations,” in *Stochastic Differential Systems, Stochastic Control Theory and Their Applications* (FLEMING, W. and LIONS, P. L., eds.), vol. 10 of *The IMA volumes in mathematics and its applications*, (New York), pp. 147–186, Springer, 1988.
- [34] HARRISON, J. M. and VAN MIEGHEM, J. A., “Dynamic control of Brownian networks: state space collapse and equivalent workload formulations,” *Annals of Applied Probability*, vol. 7, pp. 747–771, 1997.
- [35] HARRISON, J. M., “Heavy traffic analysis of a system with parallel servers: asymptotic analysis of discrete-review policies,” *Annals of Applied Probability*, vol. 8, pp. 822–848, 1998.
- [36] HARRISON, J. M., “Brownian models of open processing networks: canonical representation of workload,” *Annals of Applied Probability*, vol. 10, pp. 75–103, 2000.
- [37] HARRISON, J. M., “Stochastic networks and activity analysis,” in *Analytic Methods in Applied Probability* (SUHOV, Y., ed.), In Memory of Fridrik Karpelevich, (Providence, RI), American Mathematical Society, 2002.
- [38] HARRISON, J. M., “A broader view of Brownian networks,” *Annals of Applied Probability*, vol. 13, pp. 1119–1150, 2003.
- [39] HARRISON, J. M. and LÓPEZ, M. J., “Heavy traffic resource pooling in parallel-server systems,” *Queueing Systems: Theory and Applications*, vol. 33, pp. 339–368, 1999.
- [40] HARRISON, J. M. and NGUYEN, V., “Brownian models of multiclass queueing networks: Current status and open problems,” *Queueing Systems: Theory and Applications*, vol. 13, pp. 5–40, 1993.
- [41] IGLEHART, D. L. and WHITT, W., “Multiple channel queues in heavy traffic I,” *Adv. Appl. Probab.*, vol. 2, pp. 150–177, 1970.
- [42] KELLY, F. P. and LAWS, C. N., “Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling,” *Queueing Systems: Theory and Applications*, vol. 13, pp. 47–86, 1993.
- [43] KOOPMANS, T. C., ed., *Activity Analysis of Production and Allocation*. New York: John Wiley and Sons, 1951.
- [44] KUMAR, S. and KUMAR, P. R., “Performance bounds for queueing networks and scheduling policies,” *IEEE Transactions on Automatic Control*, vol. AC-39, pp. 1600–1611, 1994.

- [45] LABOVITZ, C., AHUJA, A., WATTENHOFER, R., and VENKATACHARY, S., “The impact of Internet policy and topology on delayed routing convergence,” *IEEE INFOCOM*, pp. 71–79, 2001.
- [46] LAWS, C. N., “Resource pooling in queueing networks with dynamic routing,” *Adv. Appl. Probab.*, vol. 24, pp. 699–726, 1992.
- [47] LAWS, C. N. and LOUTH, G. M., “Dynamic scheduling of a four-station queueing networks,” *Prob. Eng. Inf. Sci.*, vol. 4, pp. 131–156, 1990.
- [48] LU, S. C. H., RAMASWAMY, D., and KUMAR, P. R., “Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, pp. 374–385, 1994.
- [49] MANDELBAUM, A. and STOLYAR, A. L., “Scheduling flexible servers with convex delay costs: heavy traffic optimality of the generalized $c\mu$ -rule,” *Operations Research*, vol. 52, pp. 836–855, 2004.
- [50] MANGASARIAN, O. L., “Uniqueness of solution in linear programming,” *Linear Algebra and its Applications*, vol. 25, pp. 151–162, 1979.
- [51] MARSAN, M. A., GIACCONE, P., LEONARDI, E., and NERI, F., “On the stability of local scheduling policies in networks of packet switches with input queues,” *Journal on Selected Areas in Communications “High-performance electronic switches/routers for high-speed internet”*, vol. 21, pp. 642–655, 2003.
- [52] MCKEOWN, N., *Scheduling Algorithms for Input-Queued Cell Switches*. PhD thesis, University of California, 1995.
- [53] MCKEOWN, N., “iSLIP: A scheduling algorithm for input-queued switches,” *IEEE Transactions on Networking*, vol. 7, pp. 188–201, 1999.
- [54] MCKEOWN, N., MEKKITTIKUL, A., ANANTHARAM, V., and WALRAND, J., “Achieving 100% throughput in an input-queued switch,” *IEEE Transactions on Communications*, vol. 47, pp. 1260–1267, 1999.
- [55] MEYN, S., “Stability and optimization of queueing networks and their fluid models,” in *Mathematics of Stochastic Manufacturing Systems (Williamsburg, VA, 1996)*, vol. 33 of *Lectures in Applied Mathematics*, pp. 175–199, Providence, RI: American Mathematical Society, 1997.
- [56] REIMAN, M. I., “The heavy traffic diffusion approximation for sojourn times in Jackson networks,” in *Applied probability-computer science: The interface*, vol. 2, pp. 409–422, 1982.
- [57] REIMAN, M. I., “Open queueing networks in heavy traffic,” *Mathematics of Operations Research*, vol. 9, pp. 441–458, 1984.
- [58] REIMAN, M. I., “Some diffusion approximations with state space collapse,” in *Modeling and Performance Evaluation Methodology* (BACCELLI, F. and FAYOLLE, G., eds.), pp. 209–240, Berlin: Springer, 1984.
- [59] ROYDEN, H. L., *Real analysis*. New York: Prentice Hall, 3rd ed., 1988.

- [60] RUDIN, W., *Functional Analysis*. McGraw-Hill, second ed., 1991.
- [61] SEIDMAN, T. I., “‘First come, first served’ can be unstable!,” *IEEE Transactions on Automatic Control*, vol. 39, pp. 2166–2171, 1994.
- [62] SQUILLANTE, M. S., XIA, C. H., YAO, D. D., and ZHANG, L., “Threshold-based priority policies for parallel-server systems with affinity scheduling,” in *Proceedings of the IEEE American Control Conference*, pp. 2992–2999, 2001.
- [63] STOLYAR, A. L., “On the stability of multiclass queueing networks: a relaxed sufficient condition via limiting fluid processes,” *Markov Processes and Related Fields*, vol. 1, pp. 491–512, 1995.
- [64] STOLYAR, A. L., “Maxweight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic,” *Annals of Applied Probability*, vol. 14, pp. 1–53, 2004.
- [65] TASSIULAS, L., “Adaptive back-pressure congestion control based on local information,” *IEEE Transactions on Automatic Control*, vol. 40, pp. 236–250, 1995.
- [66] TASSIULAS, L. and BHATTACHARYA, P. B., “Allocation of interdependent resources for maximal throughput,” *Stochastic Models*, vol. 16, pp. 27–48, 2000.
- [67] TASSIULAS, L. and EPHREMIDES, A., “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.
- [68] TASSIULAS, L. and EPHREMIDES, A., “Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Transactions on Information Theory*, vol. 39, pp. 466–478, 1993.
- [69] THOMAS, L. J. and MCCLAIN, J. O., “An overview of production planning,” in *Logistics of Production and Inventory* (GRAVES, A. H. and ZIPKIN, P., eds.), vol. 4 of *Handbooks in operations research and management science*, North-Holland, 1993.
- [70] VAN VUREN, T. and SMART, M. B., “Route guidance and road pricing - problems, practicalities and possibilities,” *Transport Reviews*, vol. 10, pp. 269–283, 1990.
- [71] WEIN, L. M., “Scheduling networks of queues: Heavy traffic analysis of a multistation network with controllable inputs,” *Operations Research*, vol. 40, pp. S312–S334, 1992.
- [72] WEIN, L. M., “Scheduling semiconductor wafer fabrication,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 1, pp. 115–130, 1988.
- [73] WILLIAMS, R. J., “On the approximation of queueing networks in heavy traffic,” in *Stochastic Networks: Theory and Applications* (KELLY, F. P., ZACHARY, S., and ZIEDINS, I., eds.), Royal Statistical Society, Oxford University Press, 1996.
- [74] WILLIAMS, R. J., “An invariance principle for semimartingale reflecting Brownian motions in an orthant,” *Queueing Systems: Theory and Applications*, vol. 30, pp. 5–25, 1998.

- [75] WILLIAMS, R. J., “Diffusion approximations for open multiclass queueing networks: sufficient conditions involving state space collapse,” *Queueing Systems: Theory and Applications*, vol. 30, pp. 27–88, 1998.
- [76] WILLIAMS, R. J., “On dynamic scheduling of a parallel server system with complete resource pooling,” in *Analysis of Communication Networks: Call Centres, Traffic and Performance* (MCDONALD, D. R. and TUNERS, S. R. E., eds.), vol. 8 of *Fields Institute Communications*, pp. 49–71, American Mathematical Society, 2000.
- [77] WRIGHT, S. J., *Primal-Dual Interior-Point Methods*. Philadelphia, PA.: SIAM, 1997.